



**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

---

**Fakulta elektrotechnická  
Katedra radioelektroniky**

**Technologie pro streaming videa**

**Video streaming technologies**

Diplomová práce

Studijní program: Komunikace, multimédia a elektronika (magisterský)  
Studijní obor: Multimediální technika

Vedoucí práce: Ing. Karel Fliegel, Ph.D.

**Bc. Václav Popelka**

---

**Praha 2013**

# Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze 29. prosince 2013

\_\_\_\_\_  
Podpis studenta

# Zadání (přepis)

*České vysoké učení technické v Praze*

*Fakulta elektrotechnická*

*Katedra radioelektroniky*

## ZADÁNÍ DIPLOMOVÉ PRÁCE

*Student: Bc. Václav Popelka*

*Studijní program: Komunikace, multimédia a elektronika (magisterský)*

*Obor: Multimediální technika*

*Název tématu: **Technologie pro streaming videa / Video streaming technologies***

*Pokyny pro vypracování:*

*Podajte přehled současných technologií pro streaming videa. Zaměřte se zejména na nejnovější standard MPEG-DASH (Dynamic Adaptive Streaming over HTTP). Popište typické artefakty, které při streamingu videa pomocí různých technologií vznikají a analyzujte jejich vliv na vnímanou kvalitu obrazu. Realizujte model streamingového serveru a s jeho využitím porovnejte účinnost a výpočetní náročnost jednotlivých technologií.*

*Seznam odborné literatury:*

*[1] Wu, H. R., Rao, K. R.: Digital Video Image Quality and Perceptual Coding, Signal Processing and Communications, CRC Press, New York, 2006.*

*[2] Winkler, S.: Digital Video Quality: Vision Models and Metrics, Wiley, 2005.*

*Vedoucí: Ing. Karel Fliegel, Ph.D.*

*Platnost zadání: do konce zimního semestru 2014/2015*

*Prof. Ing. Miloš Klíma, CSc.*

*vedoucí katedry*

*Prof. Ing. Pavel Ripka, CSc.*

*děkan*

*V Praze dne 13.9.2013*

# Abstrakt

Cílem této diplomové práce je seznámení se se streamingovými technologiemi a jejich využití v praxi se zaměřením na nejnovější standardy. V teoretické části jsou popsány videokompresní techniky a formáty, dále pak metody doručování streamingu, nejpoužívanější aplikace pro streamování, typické artefakty, které při zpracování videa vznikají a jsou zde rovněž popsány metody pro hodnocení kvality videa. Praktická část této diplomové práce je věnována částečně porovnání kvality testovaných videosekvencí, ale hlavně streamingovým testům tří rozdílných technologií, u kterých je sledována zejména zátěž jednotlivých technologií vůči odbavovacímu serveru. Následně jsou dosažené výsledky vyhodnoceny.

Klíčová slova: formáty a komprese videa, streamingové aplikace, adaptivní streaming, MPEG-DASH ( adaptivní streaming pomocí http), kvalita videa, streaming v praxi.

# Abstract

The aim of this diploma thesis is to introduce streaming technologies and their use in practice with focus on the newest standards. Theoretical parts describes video compression techniques and formats, methods of streaming delivery, the most used streaming applications, typical artifacts caused by specific means of video processing, and, furthermore, methods for video quality evaluation. The practical part of this diploma thesis focuses partly on quality comparison of particular video sequences, yet mainly on streaming tests of three different technologies, in which especially the load of every individual technology in relation to clearance server and their consequent comparison and assessment is observed.

Keywords: video compression, video formats, streaming applications, adaptive streaming, MPEG-DASH (adaptive streaming over http), video quality, streaming in use.

## Poděkování

Děkuji panu Ing. Karlu Fliegelovi, Ph.D. za vedení mé diplomové práce a také za to, že mi vždy vyšel vstříc.

Dále děkuji zaměstnancům společnosti Visual Unity, kteří mi ochotně radili a pomohli mi s překódováním testovací video sekvence pro praktickou část této práce.

V neposlední řadě děkuji Gustavu Villegasovi, jenž mi ochotně poskytl informace ohledně projektu TV2Moro, a společnosti Píksel, která mi poskytla přednášku o projektu Sky.

Mé poděkování také patří Janovi Otte a Michalu Krskovi za zapůjčení techniky a pomoc při testování a také všem ostatním, kteří mě při psaní této práce podporovali.

# Obsah:

Prohlášení .....	2
Zadání (přepis).....	3
Abstrakt .....	4
Poděkování .....	5
Obsah:.....	6
Seznam obrázků.....	9
Úvod .....	10
1 Úvod do streamingu .....	12
1.1 Rozdělení streamingu. ....	12
1.2 Komprese dat .....	13
1.3 Video kompresní techniky .....	14
1.3.1 Redundance v komprimačních metodách.....	14
1.3.2 Irelevance.....	14
1.3.3 Run-Length Coding (RLC).....	14
1.3.4 Variable Length Coding (VLC).....	15
1.3.5 Spatial transform .....	15
1.3.6 Barevné podání a vzorkování .....	16
1.3.7 Kvantování .....	17
1.3.8 Odhad pohybu .....	17
2 Formáty videokomprese .....	20
2.1 MPEG-2 .....	21
2.2 MPEG-4 AVC/H.264.....	21
2.3 HEVC/H.265.....	22
2.4 Ogg Theora .....	23
3 Metody streamingu .....	25
3.1 Tradiční streaming .....	25
3.2 Progresivní download .....	26
3.3 HTTP-Based Adaptive Streaming .....	26
3.4 MPEG-DASH .....	27
3.4.1 MPD .....	29
3.4.2 Segmentové formáty a typy.....	31
3.4.3 Rozdělení obsahu.....	31
3.4.4 Praktické využití.....	32
4 Aplikace pro streaming.....	33
4.1 Real Media a Helix Server.....	33
4.1.1 Helix Server .....	33
4.1.2 Helix Broadcaster .....	35
4.2 Windows Media Services .....	35
4.2.1 Windows Media Services .....	36
4.2.2 Windows Media Encoder .....	37
4.3 Microsoft Smooth Streaming.....	37
4.3.1 Microsoft IIS Media Services.....	38
4.3.2 Expression Encoder .....	39
4.4 Quick Time Streaming Server .....	40
4.5 Quick Time Broadcaster .....	40
4.6 HTTP Live Streaming a Darwin Streaming Server .....	41

4.7	Adobe Flash Video .....	42
4.8	Wowza .....	43
4.9	VideoLan VLS a VLC .....	44
5	Streamovací řetězec .....	46
5.1	Zdroj videa .....	46
5.1.1	Signály DVB-T/2, DVB-S/2, DVB-C .....	47
5.1.2	Hardwarové karty pro přijímání signálů .....	47
5.2	Úprava a kódování videa .....	48
5.2.1	FFmpeg .....	48
5.3	Streamingový server - Origin .....	49
5.4	CDN .....	49
5.5	Koncový uživatel .....	50
6	Typické artefakty .....	51
6.1	Kompresní artefakty .....	51
6.2	Chyby způsobené přenosem .....	52
6.3	Ostatní poruchy .....	53
7	Hodnocení kvality videa .....	54
7.1	Subjektivní hodnocení .....	54
7.2	Objektivní hodnocení .....	55
7.2.1	MSE a PSNR .....	55
7.2.2	Měření kvality pomocí strukturálního zkreslení .....	55
7.2.3	Měření s redukovanou referencí .....	56
7.2.4	Měření bez reference .....	56
8	Streaming v praxi .....	57
8.1	iVysílání české televize .....	57
8.2	TV2Moro .....	58
8.3	Sky Go .....	59
9	Praktické testy .....	60
9.1	Cíl testování .....	60
9.2	Vybrané řešení streamingového serveru .....	60
9.3	Testovací prostředí .....	60
9.4	Testovací oblasti .....	60
9.5	Testované technologie .....	61
9.5.1	RTMP .....	61
9.5.2	RTSP .....	62
9.5.3	HDS .....	62
9.6	Metodika testování .....	62
9.6.1	JConsole .....	63
9.6.2	WMSPanel .....	64
9.6.3	Darkstat .....	64
9.6.4	Top .....	64
9.7	Testování .....	65
9.7.1	RTMP .....	65
9.7.2	RTSP .....	65
9.7.3	HDS .....	66
9.8	Srovnání .....	67
9.8.1	Grafické zpracování .....	67
9.9	Vyhodnocení výsledků .....	72
10	Závěr .....	73
11	Použité zkratky .....	74

12	Seznam použité literatury a elektronických zdrojů .....	75
13	Přílohy .....	81
13.1	Příloha 1 – použitý script pro testování.....	81
13.2	Příloha 2 – tabulky, zpracované výsledky.....	83
13.3	Příloha 3 – DVD.....	86



## Seznam obrázků

- Obr. 1.1      Příklad kvantizační matice
- Obr. 1.2      Ilustrace metody odhadu pohybu
- Obr. 1.3      Příklad sekvence GOP v MPEG-2
- Obr. 2.1      Rozdílná velikost formátů H.264 a H.265
- Obr. 3.1      Příklad Media Data Presentation modelu
- Obr. 6.1      Příklad přenosové cesty videa při streamingu
- Obr. 7.1      Příklad DSCQS metody (vlevo sekvence obrazů, vpravo hodnocení kvality)
- Obr. 7.2      DSIS metoda (vlevo sekvence obrazů, vpravo hodnocení kvality)
- Graf 9.1      Graf závislosti vytížení CPU na počtu klientů a datovém toku
- Graf 9.2      Graf závislosti vytížení RAM na počtu klientů a datovém toku
- Graf 9.3      Graf vytížení CPU na datovém toku
- Graf 9.4      Graf vytížení CPU na datovém toku

# Úvod

Dnešní dobu můžeme bez nadsázky nazvat „digitální dobou“, protože analogové přístroje jsou ve většině použití nahrazovány svými mladšími digitálními sourozenci. Zatím nemusíme kvůli možnosti prohlédnutí analogové techniky navštěvovat muzea, ale v profesionální technice již analogové přístroje, jako jsou videokamery, případně další zařízení z řetězce na pořizování, zpracování, distribuci, sledování obsahu videa či obrazu, nenajdeme.

Mnozí z nás by si tyto digitální technologie neuměli ještě před 20 lety představit, tedy v době, kdy se teprve začal internet rozšiřovat a málokdo vlastně věděl jaký potenciál internet skýtá. Dnes to bereme téměř jako samozřejmost, že si na různých zařízeních (počítače, notebooky, laptopy, netbooky, tablety či dokonce mobilní telefony) pustíme online video či hudbu.

A právě o tomto tématem, tzv. streamingem se budu zabývat podrobněji v této práci, která popisuje celý tento řetězec až po koncového uživatele.

V první části této práce jsou popsány hlavně kompresní mechanismy, které se významně podílejí na výsledné kvalitě a také na velikosti datových toků jednotlivých videí.

Druhá část práce je věnována zejména nepoužívanějším kodekům, bez nichž se streaming neobejde. Jsou zde popsány nepoužívanější kodeky dnešní doby, bez kterých by se dnešní audiovizuální technika neobešla.

Následující část práce je věnována různým metodám streamingu, jsou v ní podrobně popsány tři základní techniky doručování spolu se čtvrtou – nejnovější technologií adaptivního streamingu pomocí http protokolu.

Ve čtvrté části práce najdeme seznámení se s nejčastějšími, a v dnešní době nepoužívanějšími, streamingovými aplikacemi. Tyto aplikace v závislosti na požadované technologii jsou využívány pro doručování video obsahu.

Další část práce se věnuje streamovacímu řetězci, jsou v ní popsány možné alternativy zdroje signálu, přes jeho zpracování a využití doručovacích serverů až po koncového uživatele.

Následně se další dvě části práce zabývají typickými artefakty, ke kterým při zpracování videa dochází a hodnocením výsledné kvality videa.

Osmá část práce se zabývá praktickým využitím s příklady streamingu.

Významnou částí práce je devátá část, v níž najdeme popis praktických testů, které byly v rámci této práce prováděny tak, abychom mohli porovnat výpočetní náročnosti na tyto

technologie. Najdeme test tří technologií jenž jsou používány ke streamování, a metody, jakým způsobem byly testovány.

Závěr této práce se věnuje zpracováním, srovnáním a vyhodnocením výsledků, jsou v něm uvedeny grafy a také slovní popis dosažených výsledků.

# 1 Úvod do streamingu

Nejdříve je potřeba stručně vysvětlit, co si pod pojmem streaming máme představovat. Streaming pochází z anglického slova „stream“, což v překladu znamená proud nebo proudování.

Streaming si tedy můžeme představit jako datový tok, kterým ke koncovým klientům proudí různé audiovizuální záznamy. Za streaming se samozřejmě považuje i přenos samotného zvuku (audia), ale tato práce se zabývá spíše přenosem kompletního audiovizuálního obsahu.

Na začátku takového nejjednoduššího streamovacího řetězce může být videokamera či jiný zdroj videa, počítač<sup>1</sup> a síťové připojení, na klientské straně multimediální zobrazovací zařízení (počítač, mobil tablet apod.), kterým je možno streamovaný obsah sledovat třeba i několik set tisíc kilometrů daleko. [3]

## 1.1 Rozdělení streamingu.

Streaming je možno rozdělit podle úhlu pohledu do jednotlivých kategorií z hlediska:

- obsahu
- času
- proměnnosti bitového toku

Z hlediska obsahu se dá streaming rozdělit do dvou kategorií, kdy v první je předmětem přenosu pouze audio, a ve druhém případě audiovizuální obsah.

První kategorie by se ve skutečnosti dala charakterizovat spíše jako podskupina, ale pro lepší přehlednost ji bereme jako samostatnou kategorii. Využití má hlavně v šíření rádiového vysílání, ve chvílích či zeměpisných polohách, kdy nemáme možnost příjmu těchto rádiových vln, na kterých konkrétní rádiová stanice běžně vysílá.

Přenos kompletního audia a videa pomocí internetu z různých půjčoven, televizních stanic nebo amatérských videí však převažuje a v dnešní době zaujímá více jak 58 % z celkového přenosu dat v rámci internetu. V roce 2017 se předpokládá že to bude až okolo 70 %.<sup>2</sup>[24]

Z časového hlediska lze streaming opět dělit na dvě skupiny - podle toho, zda se na video klienti mohou dívat pouze v době vysílání nebo kdykoliv se jim to hodí. První skupinu

---

<sup>1</sup> Přesněji řečeno server s odpovídajícím hardwarovým a softwarovým vybavením.

<sup>2</sup> Údaje platí pro přenos dat v rámci statistik pro severní Ameriku. Oficiální a podložené statistiky pro celý svět jsem bohužel z dostupných zdrojů nedohledal.

nazýváme online streaming (podobně jako online televizní přenos) a druhou VoD (Video on Demand) neboli offline streaming. [4]

Z hlediska proměnnosti bitového toku budeme rozlišovat neadaptivní a adaptivní streaming.

Neadaptivní streaming je spíše na ústupu a v dnešní době se již využívá pouze v audio streamingu. V praxi neadaptivní streaming znamená, že si klient při výběrání obsahu zvolí také kvalitu ve které chce obsah sledovat. Kvalita je většinou identifikována bitovým tokem, případně pomocí rozlišení. Vyšší rozlišení, případně bitový tok, znamená kromě vyšší kvality zpravidla i vyšší náročnost na propustnost datové linky.

V některých případech si přehrávač automaticky otestuje propustnost našeho připojení a poté patřičnou kvalitu zvolí sám. Podstatou neadaptivního streamingu je, že jakmile jednou streamování začne, v jeho průběhu se kvalita již nemění.

Oproti tomu adaptivní streaming využívá aktuální propustnosti datové linky a sám v případě snížení datové propustnosti sníží i kvalitu videa (pokud je dostupná) tak, aby se stream pokud možno nezastavil. To znamená, že při dostatečně kvalitním připojení se klient může dívat na video v nejvyšší kvalitě, ale po chvíli, když někdo na stejné síti začne stahovat soubory a omezí tím propustnost, přehrávač situaci vyhodnotí a sám přepne na nižší kvalitu tak, aby se data stihla stahovat průběžně a klient mohl sledovat obsah plynule a bez výpadku.

## 1.2 Komprese dat

Streamování jako takové není jednoduchá záležitost a hlavně kvůli plynulejšímu přenosu videa používáme dva základní principy pro lepší spokojenost koncového klienta.

Jedná se zejména o různé komprese videa, kterými se budeme zabývat v následující kapitole, nebo dále o zmenšení datového toku pomocí použití nižšího rozlišení výsledného videa.

Pro různé video přenosy se používají primárně různá rozlišení s rozdílným stupněm a druhem komprese. To, co se může zdát jako dostatečné při přenosu konference, nemusí vůbec vyhovovat potřebám pro přenos rychle se pohybujících objektů při přenosu sportovního utkání.

## 1.3 Video kompresní techniky

Vzhledem k vysoké náročnosti dnes již poměrně rozšířeného formátu full HD videa (1920x1080 pixelů) je zcela zřejmé, že datový tok potřebný pro přenos tak velkého objemu dat, je potřeba nějakým způsobem zmenšit.

Nejen pro streaming, ale také pro uchovávání videa, se v praxi používají tzv. kodeky. Toto slovo vzniklo ze složení dvou anglických slov COder + DECoder a následným počestěním. V české literatuře můžeme tento výraz běžně vidět i nepřeložen jako codec.

Kodek je tedy něco, co za pomoci určitých algoritmů, dané video zakóduje a tím zmenší výslednou velikost souboru či datového toku.

Cílem kompresních technik je oddělit redundatní a irelevantní informace z digitálních dat. Současně se počítá s limitací procesu snižování objemu dat tak, aby byl výsledek pro lidské oko co nejpodobnější originálu, v ideálním případě od něj nerozpoznatelný.

Lidským okem zde není myšleno pouze oko jako takové, ale spíše celý zrakový systém označovaný také jako HVS - Human Visual System.

### 1.3.1 Redundance v komprimačních metodách

Redundance je bezztrátovou (lossless) metodou komprimace. To znamená, že zrekonstruovaný soubor dat je identický s jeho originálem. Tyto bezztrátové metody využívají nahrazování binárních sekvencí bloky s více kompaktními reprezentacemi. [5]

### 1.3.2 Irelevance

Irelevance je v podstatě část informace, kterou díky omezenosti lidského vnímání nepostřehneme, informace pro lidské oko bezvýznamná. Proto jsou všechny informace, které jsou pokládány za „nadbytečné“ a lidským okem nepostřehnutelné, odděleny. Po odstranění irelevance nemůže být výsledný signál stejný jako signál původní, tudíž dochází ke ztrátě části informací, což vede ke ztrátové kompresi.

### 1.3.3 Run-Length Coding (RLC)

RLC je jedna z nejjednodušších metod pro komprimaci dat. Pracuje na základě využití principu opakování stejných symbolů dat, přičemž je vhodná spíše pro souvislý datový tok.

Bylo vyvinuto více možností realizace RLC metody. Například podle rozdělení vstupních dat do skupin podle jednotlivých bajtů.

V MPEG kompresním algoritmu je tato metoda mírně upravena tak, že RLC blok dat obsahuje větší množství nulových symbolů. Proto je složena ze dvou hodnot, z nichž první reprezentuje nulovou hodnotu koeficientu před dalším nenulovým koeficientem, který je druhý v řadě.

### 1.3.4 Variable Length Coding (VLC)

VLC pracuje na principu určování statistické četnosti datových symbolů, jenž zastupuje proměnnou délkou kódu. Tato délka zastupujícího kódu závisí na pravděpodobnosti výskytu v určitém bitovém toku, kdy jsou nejčastější symboly dat reprezentovány nejkratším kódem a naopak. Například v prostém textu bývá nejčastěji zastoupen symbol mezeru – který se tak nahrazuje nejkratší možnou bitovou reprezentací. Nejznámější metodou VLC je Huffmanovo kódování (například ve standardu MPEG). [5]

### 1.3.5 Spatial transform

Spatial transform (prostorová transformace) oproti RLC a VLC používá čistě statistickou metodu ke snížení počtu bitů. Převádí vstupní blok obrazových hodnot na ekvivalentní zastoupení ve frekvenční oblasti, protože lidské vnímání je vysoce závislé na vnímání prostorové frekvence. Díky tomu můžeme eliminovat vysoké prostorové frekvenční kmitočty při zachování vysoké kvality pro lidské oko. Nejčastěji používanou metodou je diskrétní kosínova transformace (Discrete Cosine Transform, DCT). Další používanou metodou je diskrétní vlnková transformace (Discrete Wavelet Transform, DWT).

DCT vychází z diskrétní Fourierovy transformace (Discrete Fourier Transform, DFT), a je základem mnoha kompresních algoritmů – například JPEG, MPEG-1/2/4, H.261/3/4 nebo Ogg Theora. DCT je nejčastěji používána na blocích 8x8, pro kterou jsou upraveny i následující vzorce, přičemž  $i, j$  jsou souřadnice v kmitočtové oblasti a  $x, y$  souřadnice v časové oblasti. [1], [6], [7].

DCT pro pole 8x8 je definována jako [27]:

$$F[u, v] = \frac{1}{4} C(u)C(v) \sum_{i=0}^7 \sum_{j=0}^7 \left( f[i, j] \cos\left(\pi u \frac{(2i+1)}{16}\right) \cos\left(\pi v \frac{(2j+1)}{16}\right) \right) \quad (1)$$

Následně pak inverzní IDCT je definována jako

$$f[i, j] = C(u)C(v) \sum_{u=0}^7 \sum_{v=0}^7 \left( f[u, v] \cos\left(\pi u \frac{(2y+1)}{16}\right) \cos\left(\pi v \frac{(2x+1)}{16}\right) \right) \quad (2)$$

$$\text{kde } C(u) = \frac{1}{\sqrt{2}} \text{ pro } u=0, C(u)=1 \text{ jinak}$$

$$C(v) = \frac{1}{\sqrt{2}} \text{ pro } v=0, C(v)=1$$

### 1.3.6 Barevné podání a vzorkování

Barvy jsou digitálně reprezentovány nejčastěji rozkladem na tři základní barevné složky a to červenou, zelenou a modrou, tento model je znám jako RGB (Red, Green, Blue). Tyto barvy byly vybrány s ohledem na lidské vnímání. Vzhledem k objemnému vyjádření těchto hodnot (při 8-bitovém vyjádření jedné barvy je potřeba 24 bitů na jeden pixel) se RGB v digitálních systémech nepoužívá, a je většinou nahrazena systémem  $Y C_R C_B$  dle doporučení ITU.R BT601. Následná rovnice pak vypadá takto:

$$\begin{aligned} Y &= 0,587G + 0,299R + 0,114B \\ C_R &= R - Y \\ C_B &= B - Y \end{aligned} \quad (3)$$

kde Y znamená jasovou složku (luminance) a C znamená chrominanční složku.

Jelikož má lidské oko méně barevných fotoreceptorů pro obdržení informace o barevném podání scény, lidský zrakový aparát je mnohem méně citlivější na vnímání barevných rozdílů než na vnímání jasových rozdílů.<sup>3</sup>

Díky tomu je možno během procesu digitalizace přistoupit ke zredukování barevných složek, aniž by lidské oko zjistilo výrazné zhoršení kvality vnímaného signálu.

V zásadě se používají čtyři nejčastější formáty s označením poměru kódování 4:4:4, 4:2:2, 4:2:0 a 4:1:1, kde první číslo vyjadřuje jasovou složku, druhá dvě čísla pak barevné signály R-Y ( $C_R$ ) a B-Y ( $C_B$ ). V případě použití formátů 4:1:1 a 4:2:0 je tak ušetřena třetina dat oproti formátu 4:4:4. [1]

<sup>3</sup> Lidské oko má na sítnici dva druhy receptorů: tyčinky a čípky, které se dále dělí na 3 skupny podle barev, které zachycují: L, M a S-čípky odpovídající vlnovým délkám na které mají největší citlivost. Tzn. že reagují nejvíce na barvy červenou, zelenou a modrou.



### 1.3.7 Kvantování

Kvantování je jednoduchá metoda pro kompresi DCT koeficientů s regulovaným množstvím ztráty informací. Tento proces snižuje počet bitů potřebných pro zastoupení každého koeficientu tak, že vydělí DC koeficienty odpovídajícími hodnotami kvantizační maticí. Řada matic byla navržena tak, že každý koeficient je navržen relativně vzhledem k jeho vlivu na lidské vnímání obrazu.

Nejvyšší ztráta DC koeficientů je v pravém dolním rohu matice, kde se nacházejí vyšší frekvence. Optimální kvantizační matice definována skupinou JPEG je uvedena na následujícím obrázku, kde QM vyjadřuje střední hodnotu (může nabývat 0-100). Při zvýšení dostaneme kvalitnější obrázek s nižším kompresním poměrem, při snížení je tomu naopak. Uvedené hodnoty v kvantizační tabulce se následně přepočítávají dle níže uvedených vzorců. [8]

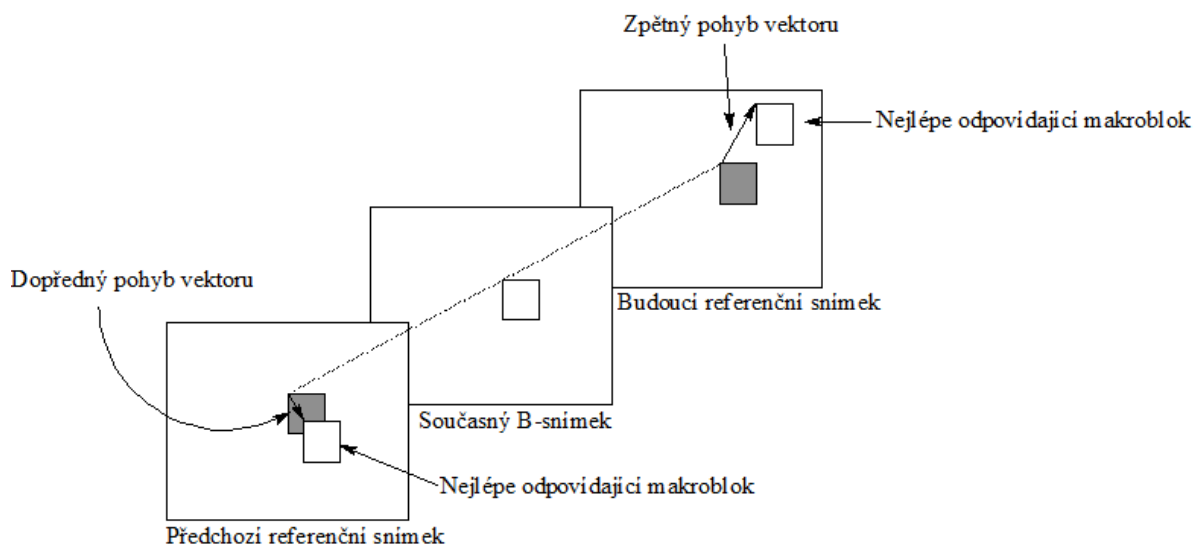
$$QM_{50} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Obrázek 1.1: Příklad kvantizační matice [8]

$$Q_q = \frac{Q_{50}(100-q)}{50} \text{ pro } Q_q > 50, \quad Q_q = \frac{50Q_{50}}{q} \text{ pro } Q_q < 50 \quad (4)$$

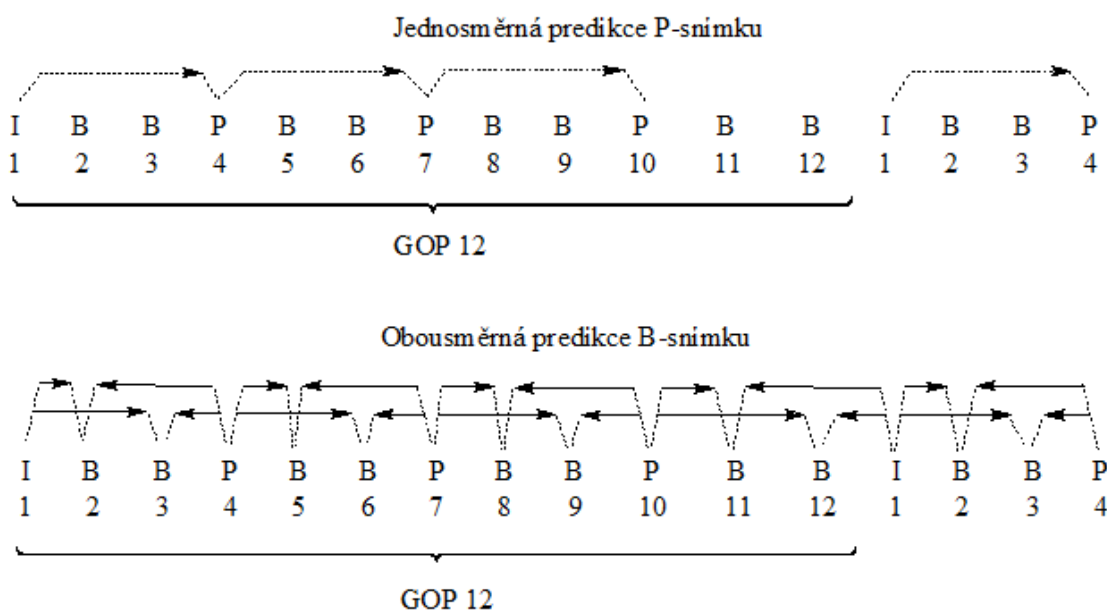
### 1.3.8 Odhad pohybu

Odhad pohybu je další účinnou technikou pro kompresi videa. Tento proces je založen na velmi malém rozdílu mezi dvěma po sobě následujícími snímky, kdy se určitá skupina bodů zobrazující předmět pohybuje na scéně ať už pohybem kamery či pohybem předmětu samotného.



Obrázek 1.2: Ilustrace metody odhadu pohybu [27]

Během procesu odhadu pohybu algoritmus hledá ve vymezené oblasti v následujícím snímku stejné makrobloky (makrobloky s odpovídající jasovou a barevnou maticí) jako byly ve snímku předchozím. Toto se provádí za pomoci střední kvadratické odchylky MSE (Mean Square Error) mezi původním makroblokem a všemi okolními makrobloky ve snímku následujícím. MSE s nejnižší hodnotou je pak brán jako přestěhovaný makroblok a je tedy reprezentován vektorem pohybu. To znamená, že není potřeba uchovávat informaci o makroblocích v každém snímku, ale stačí uchovat informaci o vektoru pohybu.



Obrázek 1.3: Příklad sekvence GOP v MPEG-2 [27]

Na obrázku 1.3 je znázorněna sekvence obrázku, která je vytvořena během kódování v MPEG-2 kompresi. Tato sekvence se nazývá GOP (Group Of Pictures) a je složena ze tří typů snímků: I-snímek, P-snímek a B-snímek. I-snímky (intra frames) obsahují kompletní informaci o celé scéně a slouží také jako referenční snímky pro predikační mechanismy, P-snímky (forward predicted frames) obsahují předpokládaný pohyb makrobloků z předchozího snímku a B-snímky (bi-directionally predicted frames) obsahují pouze informace o rozdílech mezi předchozím a následujícím snímkem. [1], [27]

## 2 Formáty videokomprese

Během posledních dvou desetiletí bylo vyvinuto mnoho kompresních formátů. Řada z nich byla následně překonána novějšími a lepšími, z některých se staly standardy. Zde je přehled těch nejpoužívanějších:

- MPEG-2 - standard z roku 1994 pro DVB, DVD a mnoho profesionálních broadcastových aplikací [10]
- ITU-T H.263 – nejčastěji používán pro videokonference nebo 3GPP sítích [13]
- MPEG-4 ASP – používán nejvíce pro dohledové systémy a IP kamerové aplikace [11]
- Windows Media Video (Basic Profile) - proprietární kompresní formát, který byl hojně využíván pro internetový streaming (dnes je již na ústupu). Tento formát však podporuje chráněný obsah pomocí Windows Media DRM [16]
- Real Media Video – proprietární kompresní formát používaný pro internet streaming (nebo 3GPP), umožňující streamování obsahu chráněného pomocí DRM [17]
- Proprietární MPEG-4 implementovaný například v DivX nebo XviD – používán pro umístování kompletních videí.
- Ogg Theora – jako jediný je open-source, používaný pro streaming a sdílení videí. [15]
- MPEG-4 AVC/H.264 – (Advanced Video Coding) je nástupcem MPEG-2, v dnešní době hodně využíván ve většině profesionálních aplikací [12], [14]
- HEVC/H.265 – (High Efficiency Video Coding) nejnovější standard videokomprese, který umí zmenšit datový tok potřebný pro stejnou kvalitu jako předchozí H.264 až na polovinu. [25]
- Windows Media Video Advanced Profile – standardizovaný jako SMPVC-1, byl velkým konkurentem MPEG-4 AVC/H.264, ale dnes již nemá takové velké postavení na trhu.
- VP8/WebM - otevřený standard (BSD) od společnosti Google, jenž používá pro audio kompresi Vorbis audio kodek. Má podobné nároky na datové přenosy jako H.264. [26]

## 2.1 MPEG-2

Kompresní formát MPEG-2 byl zaveden v roce 1993 a stal se standardem v profesionálních broadcastových aplikacích, DVB i domácí zábavě v podobě DVD.

Blokové schéma MPEG-2 můžete vidět na obrázku 2.1. Jako první v celém procesu se obraz snímku rozdělí na bloky o velikosti 8x8 pixelů.

V případě I-snímku jsou všechny vzorky převedeny do frekvenční oblasti a následně kvantizovány. Poté se na koeficienty matice provede metoda zig-zag a na výslednou bitovou sekvenci dat se provede entropie kódování (RLC, Huffman). Konečný bitový tok je multiplexován s doplňujícími informacemi a ukládán v bufferu (zásobníku), který je součástí ovládání bitového toku a je ovládán výstupní částí kodéru. V případě P a B snímků je použita metoda odhadu pohybu. [27]

## 2.2 MPEG-4 AVC/H.264

MPEG-4 AVC-H.264 je jeden z nejnovějších standardizovaných komprimačních formátů a je využíván v profesionálních broadcastových řešeních. Ve srovnání s MPEG-2 obvykle poskytuje lepší komprimační poměr a tím může ušetřit až 50% výsledných dat při zachování stejné kvality obrazu. MPEG-4 AVC/H.264 standart byl vyvinut experty z ISO/IEC Moving Pictures Group a ITU-T Video Coding Experts Group. [11], [14], [18]

Ve srovnání s předchozími MPEG a ITU-T standardy přináší MPEG-4 AVC/H.264 řadu nových funkcí v celém procesu kódování videa:

- vylepšená metoda odhadu pohybu
  - multi-snímková predikce (umožňující porovnávání s více než jedním předchozím snímkem)
  - quarter pixel motion vector
  - makrobloky mohou být rozděleny do 7 různých typů bloků (viz obrázek 2.2)
  - každý z těchto bloků může mít svůj vlastní nezávislý vektor pohybu
  - zvýšení účinnosti kódování díky delší struktuře GOP
- mezisnímková predikce
  - predikované bloky jsou tvořeny na základě již dříve zakódovaných a zrekonstruovaných bloků pomocí jednoho z devíti režimů
- nová metoda entropie kódování

- CAVLC (Content-Adaptive Variable Length Coding) – kombinace adaptivního Huffmanova kódování a Exp-Golomb kódování.
- CABAC (Content-Adaptive Binary Arithmetic Coding) – přiřazení kódu k sekvenci symbolů.
- deblokový filtr
  - provádí vyhlazení hran na blocích tak že subjektivní vnímání obrazu je lepší
  - toto je také používáno po inverzní transformaci v kodéru, aby se zmenšily drobné nedostatky predikce.

## 2.3 HEVC/H.265

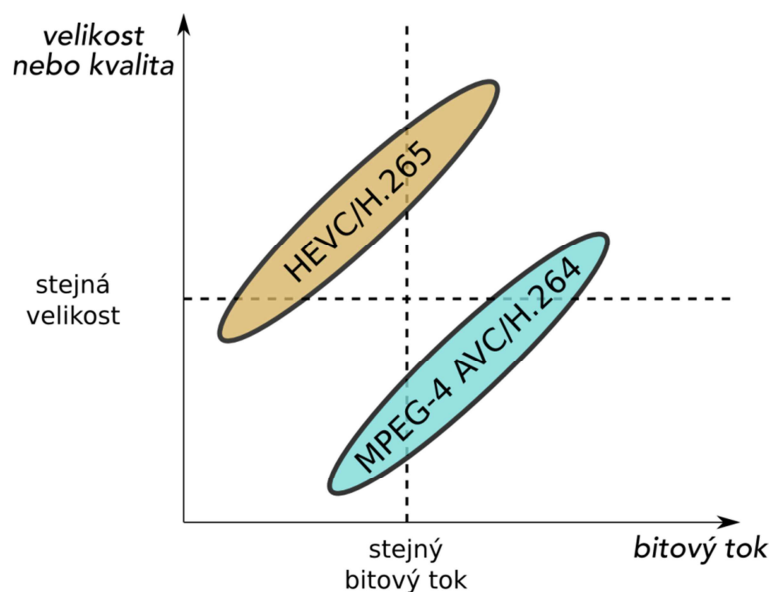
High Efficiency Video Coding (HEVC) standard je nejnovější společný projekt organizací ITU-T Video Coding Experts Group (VCEG) a ISO/IEC Moving Picture Experts Group (MPEG), které spolupracují pod pracovní skupinou s názvem Joint Collaborative Team on Video Coding (JCT-VC). Tento standard byl schválen ITU-T 13.dubna 2013 pod názvem ISO/IEC 23008-2. [28]

HEVC vychází z předchozích verzí MPEG-2 a MPEG-4 AVC/H.264, avšak s následujícími vylepšeními:

- lepší přizpůsobení rozdělení od velkých částí až po velmi malé
- větší flexibilita v predikovacích režimech a transformování velikosti bloků
- sofistikovanější interpolace a deblokový filtr
- lepší predikce a signalizace režimu pohybových vektorů
- funkce podporující účinné paralelní zpracování

Výsledkem toho je standard, který má lepší komprimační vlastnosti za cenu zvýšení výpočetní náročnosti. S HEVC je možné uložit nebo streamovat video efektivněji než při použití předchozí verze H.264.

Při stejném rozlišení a kvalitě HEVC video využívá méně místa na disku, případně potřebuje menší kapacitu přenosového pásma než jeho ekvivalent H.264, což je znázorněno na obrázku 2.1.



Obrázek 2.1: Rozdílná velikost formátů H.264 a H.265 [25]

HEVC je založeno na stejném principu jako předchozí standardy využívající diskretní kosínovou transformaci (DCT) nebo diskretní sínovou transformaci (DST) aplikovanou na bloky o velikostech 32x32, 16x16, 8x8 nebo 4x4.

Tento standard otevírá cestu k lepšímu požitku ze streamovaného videa dokonce i v UHD rozlišení (4K, případně 8K UHD). Možná v budoucnu nahradí starší verzi v digitální technice, jako digitálním televizním nebo satelitním vysílání, v oblasti přenosných DVD a Blu-Ray disků. [25], [28]

## 2.4 Ogg Theora

Ogg Theora patří mezi nepatentované a nelicencované „open-source“ kompresní formáty, které byly vymyšleny XIPH.Org Foundation. Algoritmus Theora je myšlen jako konkurent k MPEG-4 (H.264/AVC), Real Video, Windows Media Video a podobným formátům s malým datovým tokem. Vzhledem k účinnosti komprese je tento formát někde na pomezí mezi standardy MPEG-2 a MPEG-4 (H.264/AVC).

Theora byla původně založena na kodeku VP3 od On2 Technologies. Přestože VP3 je patentovaná technologie, v září 2001 byla darována veřejnosti k volnému použití. Tato skutečnost spolu s poskytnutím celé specifikace byla výhodou pro další zlepšování.

Kompresní algoritmus Theory je velmi podobný MPEG-2 standardu, ale má některá vylepšení zlepšující kompresní funkci bez porušování patentových omezení:

- DCT (Discrete Cosine Transform) s kvantováním výsledných koeficientů
- Mezisímková metoda odhadu pohybu
- Entropické kódování – VLC (Variable Length Coding) a Huffmanovo kódování. [15]



## 3 Metody streamingu

Metody streamingu se mohou lišit na základě použité technologie a přehrávače pro který je zvolen. Následuje přehled doručovacích mechanismů a používaných protokolů.

Doručování video obsahu se v dnešní době děje pomocí tří metod: tradiční streaming, progresivní download a adaptivní streaming.

Pro adaptivní streaming vznikl i standard MPEG-DASH, přiblížen v poslední sekci této kapitoly

### 3.1 Tradiční streaming

RTSP<sup>4</sup>, RTMP<sup>5</sup> a MMS<sup>6</sup> jsou vhodnými příklady tradičního streamingového protokolu, který je definován jako „stateful“ protokol, což znamená, že klient je k streamingovému serveru připojen od začátku do konce přenosu.

Tyto protokoly využívají dva druhy spojení – jeden pro doručování video obsahu, druhý pak pro kontrolní spojení mezi přehrávačem a serverem. Klient může díky kontrolnímu připojení využít funkcí PLAY, PAUSE nebo ukončit spojení.

Poté, co se vytvoří spojení mezi klientem a serverem, začne server posílat fragmenty videa v malých paketech, typicky o velikosti 1452 bytů<sup>7</sup>. V případě, že stream má datový tok 1 Mbps, jednotlivé pakety jsou přenášeny přibližně po 11 milisekundách. Tyto pakety mohou být přenášeny jak pomocí síťového protokolu UDP tak TCP, který je upřednostňován v případech blokování UDP spojení.

TCP protokol je ovšem složitější<sup>8</sup> a tak má TCP spojení typicky vyšší latenci než přenos pomocí bezstavového UDP.

HTTP je oproti předchozímu RTSP znám jako „stateless“ protokol. V případě kdy HTTP klient požaduje nějaká data, server je pošle, přičemž si nemusí pamatovat klienta a nemusí s ním udržovat stálé spojení. Každý HTTP request je odbaven jako samostatný požadavek. [35]

---

<sup>4</sup> Real Time Streaming Protocol

<sup>5</sup> Real Time Messaging Protocol

<sup>6</sup> Microsoft Media Services

<sup>7</sup> Formát těchto paketů je znám pod názvem RTP

<sup>8</sup> Zaručuje například přijetí paketů ve správném pořadí nebo nové odeslání ztraceného paketu

## 3.2 Progresivní download

Další běžně používanou formou doručování obsahu je progresivní download, který není ničím jiným než obyčejným stažením souboru z HTTP serveru.

Progresivní download je podporován velkou částí přehrávačů i platforem včetně Adobe Flash, Silverlight a Windows Media Player. Slovo „progressive“ vyplývá z toho, že většina přehrávačů je schopna přehrávat multimediální soubor, zatímco jeho stahování probíhá – t.j. předtím než je kompletní soubor zapsán na disk<sup>9</sup>. Přehrávače podporující HTTP 1.1 mohou dokonce v souboru seekovat a uživatel tak může přeskočit i na pozici, která ještě nebyla stažena. Tímto zažádá HTTP server o zaslání požadovaného rozmezí dat.

Progresivní download je v dnešní době využíván Web portály jako například YouTube, Vimeo, MySpace, MSN nebo Soapbox.

Na rozdíl od klasického streamingu, kdy se data málokdy posílají s více než 10 sekundovým předstihem, zde se data stahují do té doby, dokud není stažen celý soubor.

Nevýhodou je, že jakmile se uživatel rozhodne video v polovině zastavit a dále nesledovat, byl už pravděpodobně celý soubor stažen, přičemž byla zbytečně využita kapacita přenosového pásma ke stažení celého souboru.[35]

## 3.3 HTTP-Based Adaptive Streaming

Adaptivní streaming je jakýmsi hybridem, který funguje jako streaming, ale je založený na postupném stahování HTTP (progresivním downloadem). Tato moderní koncepce však nevyužívá nový protokol, nýbrž pracuje s protokolem HTTP, přičemž využívá dělení dlouhého souboru na mnoho velmi malých segmentů.

Při adaptivní streamingové implementaci se video/audio rozdělí na mnoho malých segmentů, tzv. „chunků“, a ty jsou terpve kódovány do daného formátu. Tyto chunky jsou dlouhé typicky od 2 do 4 sekund. Na úrovni kodeku to znamená, že se každý chunk rozdělí podle GOP tak, aby každý chunk měl klíčový snímek, bez závislostí na snímcích minulých nebo budoucích. Toto umožňuje následné dekódování každého chunku nezávisle na ostatních.

Jednotlivé chunky jsou hostovány na HTTP serveru a postupně požadovány klientem.<sup>10</sup> Kousky videa se následně stahují za sebou jako by to byly jednotlivé soubory, přičemž se použije metoda progresivního downloadu, viz výše.

---

<sup>9</sup> Typicky pouze do dočasné cache webového prohlížeče, nicméně idea zůstává platná

<sup>10</sup> Myšleno klientskou aplikací

Vzhledem k tomu, že chunky jsou zakódovány bez mezer a bez bezpřekrývání, tak je výsledný obraz pro koncového uživatele nerozeznatelný od jednoho video souboru.

„Adaptivní“ část řešení přichází v době, kdy se video kóduje v několika různých kvalitách.<sup>11</sup>

Přehrávač si tedy může volit mezi různými kvalitami.

HTTP servery totiž nejčastěji fungují na principu co nejrychlejšího doručení dat k uživateli na základě možné průchodné šířky pásma linky, přehrávač tedy jednoduše může vyhodnotit tuto propustnost a na tom základě rozhodnout, jak vysoký bitrate bude stahovat. Velikost přehrávací/stahovací paměti je plně volitelná.

Adaptivní streaming, stejně jako ostatní formy HTTP doručování, nabízí oproti tradičnímu streamování následující výhody:

- levnější nasazení, protože adaptivní streaming využívá HTTP cache/proxy a nevyžaduje specializované servery v každém uzlu sítě
- lepší dostupnost snížením problému v koncových částech sítě, kterou nemůžeme ovlivnit. Kvalita videa se může v průběhu dynamicky přepínat z horší na lepší a naopak, a tím reagovat na snížení/zvýšení propustnosti sítě.

To přináší koncovému uživateli následující výhody:

- rychlé spuštění a zkrácení času pro posouvání, protože při těchto operacích může být video spuštěno na nejnižší přenosové rychlosti s přechodem na vyšší přenosovou rychlost později, jakmile se začne přehrávat v normálním režimu
- data se nestahují do vyrovnávací paměti, uživatel není odpojen a stream se nezastavuje (pokud uživatel splňuje podmínky pro minimální datový tok)
- plynulé přepínání přenosové rychlosti založené na aktuální propustnosti sítě a vytíženosti CPU
- obecně konzistentnější a plynulejší přehrávání a tudíž lepší zážitek z obsahu.

[35]

## 3.4 MPEG-DASH

MPEG-DASH je nejnovější doručovací metodou video obsahu. Vzhledem ke svému mládí zatím není tato metoda v praxi příliš podporována, vzhledem k jejím přednostem však určitě brzy nahradí starší technologie.

---

<sup>11</sup> V různých datových tocích – tzv. bitrate

Všechny tři hlavní metody adaptivního streamingu používané v dnešní době (HLS od Apple, Smooth Streaming od Microsoftu a HDS od Adobe) jsou z větší části podobné, bohužel však nejsou 100% kompatibilní. Pro každou technologii je nutné mít speciální přehrávač, který tento formát umí zpracovat. Tato rozříštěnost v metodách doručování vedla k tomu, že v dubnu roku 2009 pracovní skupina MPEG<sup>12</sup> požádala o návrhy nového standardu. Během 3 měsíců obdržela celkem 15 návrhů, na jejichž základě potom začala spolu s organizací 3GPP<sup>13</sup> vyvíjet nový standard. Výsledkem byl dynamický adaptivní HTTP streaming MPEG DASH - Dynamic Adaptive Streaming over HTTP.

MPEG DASH není systém, protokol, kodek nebo klient, ale nový standard pro podporu efektivního a vysoce kvalitního streamingu. MPEG DASH byl standardizován na začátku roku 2012 pod normou ISO/IEC 23009-1.

Design DASH byl navržen tak, aby mohl využít:

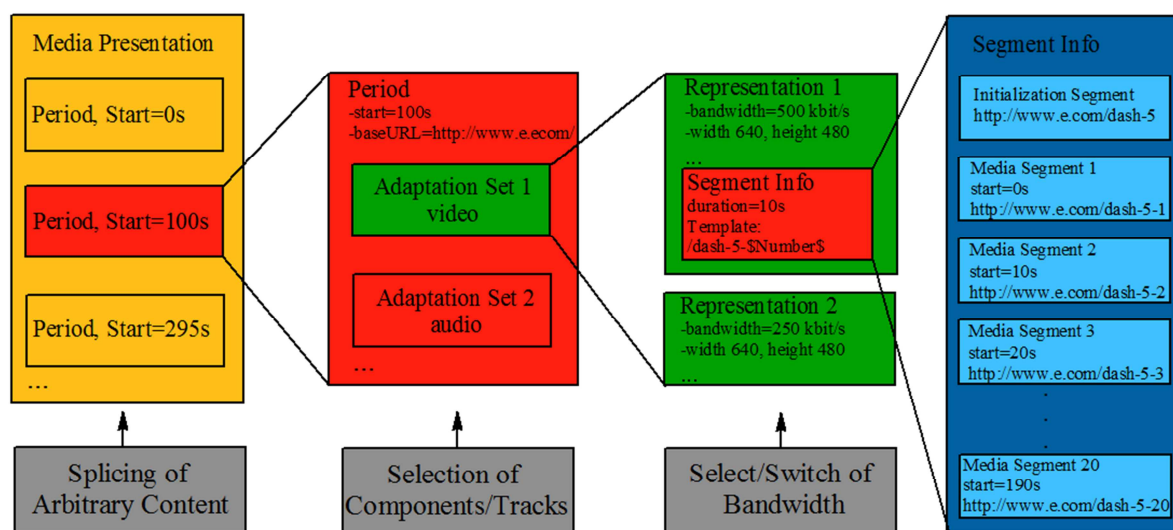
- existující technologie (video kontejnery, kodeky, DRM apod.)
- stávající síťovou infrastrukturu zaměřenou na HTTP
- uživatelské pohodlí v podobě rychlého startu a co nejmenšího čekání
- hladkého přepínání mezi kvalitami
- streaming jak live, tak VoD obsahu, případně time-shiftového obsahu
- možnosti přesunout řízení a rozhodování ze strany sítě na stranu klienta

DASH, podobně jako proprietární protokoly pro adaptivní streaming zmíněné výše, využívá k doručení obsahu fragmenty, které se doručují pomocí protokolu HTTP až ke klientovi, který je „poskládá“ v jedno souvislé video.

---

<sup>12</sup> Moving Picture Experts Group

<sup>13</sup> Third Generation Partnership



Obrázek 3.1: Příklad Media Data Presentation modelu [48]

Standard MPEG DASH stanovuje zvláštní požadavky pro využití dvou stávajících segmentových formátů MPEG-2 Transport Stream a ISO base media file format (ISO BMFF). MPEG-2 TS je segmentový formát, který je nyní využíván v HLS, zatímco ISO BMFF (v podstatě MPEG-4 formát) je využíván v HDS a Smooth Streamingu. Spojením těchto dvou formátů kontejnerů tří velkých hráčů na trhu umožňuje relativně snadnou migraci stávajícího obsahu do platformy MPEG DASH, protože segmentování většinou může zůstat stejné, jen indexové soubory musí být přeneseny do nového formátu známého jako Media Presentation Description (MPD). [46], [48], [49]

### 3.4.1 MPD

MPD, podobně jako manifest file u Smooth Streamingu (podrobněji popsany v kapitole 4.3.1), případně podobné soubory u dvou zbývajících technologií, obsahuje popis obsahu, jenž je k dispozici, včetně URL adres jednotlivých segmentů (chunků), bytových rozsahů, datového toku, rozlišení a druhu šifrování.

Po doručení Media Presentation Description souboru klientskému přehrávači je vlastní obsah (video, audio, titulky nebo data) stahován do přehrávače pomocí HTTP protokolu jako sekvence souborů, které se přehrávají v souvislém videu s možným přepínáním datových toků v závislosti na podmínkách sítě. Toto rozhodování je tedy na

klientovi. MPEG DASH nepředepisuje žádné specifické funkce, potřebné k přehrávání na straně klienta, řeší spíše formátování obsahu a přidružený MPD soubor.

Zde je ukázka rozdílů mezi HLS m3u8 a MPEG-DASH MPD soubory:

### **Index.m3u8**

```
#EXTM3U
#EXT-X-STREAM-INF : PROGRAM -ID=1, BANDWIDTH 291500, RESOLUTION 320x180
soubor1.m3u8
#EXT-X-STREAM-INF : PROGRAM -ID=1, BANDWIDTH 610560, RESOLUTION 512x288
Soubor2.m3u8
#EXT-X-STREAM-INF : PROGRAM -ID=1, BANDWIDTH 291500, RESOLUTION 1024x576
Soubor3.m3u8
```

### **Index.mpd**

```
<?xml version="1.0" encoding="utf-8"?>
<MPD
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mpeg:DASH:schema:MPD:2011"
  xsi:schemaLocation="urn:mpeg:DASH:schema:MPD:2011"
  type="static"
  mediaPresentationDuration="PT12M34.041388S"
  minBufferTime="PT10S"
  profiles="urn:mpeg:dash:profile:isoff-live:2011">

  <Period>
    <AdaptationSet
      mimeType="audio/mp4"
      segmentAlignment="0"
      lang="eng">
      <SegmentTemplate
        timescale="10000000"
        media="audio_eng=$Bandwidth$-$Time$.dash"
        initialisation=" audio_eng=$Bandwidth$.dash">
        <SegmentTimeline>
          <S t="667333" d="39473889" />
          <S t="40141222" d="40170555" />
          ...
          <S t="7527647777" d="12766111" />
        </SegmentTimeline>
```

```

</SegmentTemplate>
<Representation id="audio_eng=96000" bandwidth="96000" codecs="mp4a.40.2"
audioSamplingRate="44100" />
</AdaptationSet>
<AdaptationSet
mimeType="video/mp4"
segmentAlignment="true"
startWithSAP="1"
lang="eng">

```

### 3.4.2 Segmentové formáty a typy

Segmentový formát specifikuje syntaxi zdrojů použitých v MPD formátu a odkazuje jednoznačně na URL odkaz. Na dotazovací požadavek typu HTTP-GET na zdroj v MPD souboru je HTTP odpověď s požadovaným segmentem uvedeným v tomto souboru.

Standard definuje následující segmentové typy:

- inicializace – segmenty obsahující inicializační informace pro přístup k různým reprezentacím
- média – segmenty obsahující zakódované složky mediálního obsahu
- index – segmenty obsahující především informace o indexování Média segmentů
- bitstreamový přepínač – segmenty, v nichž jsou obsaženy základní údaje k přepnutí.

### 3.4.3 Rozdělení obsahu

U MPEG DASH se obsah může skládat z jedné nebo více částí – například zvlášť audio a video. Dlouhé položky v obsahu se mohou rozdělit do jedné nebo více dočasných kapitol, někdy nazývaných jako periody.

Dále může být každá alternativa (reprezentace) konstrukční části obsahu rozdělena do media segmentů

Klient tedy zpracovává metadata dané periody a žádá o příslušné segmenty této části. Poté dostane metadata pro další segment, na základě čehož žádá o další části.

Obecně platí že signalizační metadata DASH mohou být rozdělena do těchto kategorií:

- obsahové informace – (content-level information) zahrnují společný popis obsahu, jako jsou například čas, doba trvání, minimální vyrovnávací paměť atd.
- informace periody - (Period-level information) poskytují společný popis periody jako například čas začátku, dobu trvání, unikátní identifikátor apod.
- Informace přizpůsobení – (Adaptation information) popisují vlastnosti reprezentací a reprezentačních skupin (nazývané taky jako adaptativní skupiny v DASHi) jako například datový tok, rozlišení kvalitu.
- mapovací informace – (Mapping information) obsahují umístění pro načtení skutečného obsahu. Ve standardu DASH informace o lokalizaci nejsou pouze v MPD souboru, ale také v některých částech média segmentového formátu.

Pro efektivní řízení/zpracování metadat mohou tyto informace být fragmentovány do různých částí. Toto rozdělení umožňuje lepší efektivitu při ukládání dat a doručování. Například během relace může být obsahová informace předána pouze jednou a následně stačí pravidelné načítání periodové informace. [46], [48], [49]

### 3.4.4 Praktické využití

Prozatímni využití technologie MPEG-DASH není bohužel nijak masové. Možná je to ovlivněno pouze tím, že velcí hráči na trhu se nechtějí vzdát svého standardu, nebo třeba tím, že tato technologie je teprve novinkou a dosud není široce podporována.

Minimálně co se týče oblasti přehrávačů – ne všechny přehrávače jsou s MPEG-DASH kompatibilní, například při využití DASH technologie není možné stream z Wowzy sledovat ve VLC přehrávači, přestože nejnovější podpora VLC již MPEG-DASH podporuje.

Tato technologie je z pohledu firem zajišťující streaming určitě zajímavá, ne však už tolik z pohledu výrobců streamingových softwarů, kteří si brání svoje patenty a technologie.

Pokud se tito velcí hráči nedomluví a nepřijmou tento standard všichni společně může se stát, že DASH zůstane pouze další alternativou k metodám současným, místo aby je nahradil.

Kroky, díky nimž by se DASH stal nejpoužívanější technologií, nebudou určitě učiněny přes noc a nebude to ani v řádu několika týdnů, ale snad se k plošnému používání jednoho streamingového standardu alespoň blížíme.



## 4 Aplikace pro streaming

Za posledních pár let, kdy se zvýšila poptávka po distribuci multimediálního obsahu pomocí streamingových technologií, se snaží různé společnosti v této oblasti prosadit.

Následujících částí této kapitoly jsou věnovány nejpoužívanějším softwarovým aplikacím.

### 4.1 Real Media a Helix Server

Společnost RealNetworks byla založena v roce 1994 ve městě Washington a byla první společností v oblasti streamingu audia a videa. Jako první streamovala audio přes internet v roce 1995 - baseballový zápas. S video streamingem začala v roce 1997 a podle některých záznamů měl do roku 2000 formát Real media více než 85% podíl streamingového obsahu.

Real media formát s příponou \*.rm (případně \*.ra pro audio) umožňuje používat většinu známých komprimačních metod včetně MPEG-4 formátu.

V dnešní době tato společnost působí zejména v severní Americe, Evropě a latinské Americe prostřednictvím GameHouse, Zylom a Atrativa.

RealNetworks nabízí kompletní tzv. end-to-end streamingové řešení založené na platformě Helix Media Delivery. Do této platformy patří jak hardwarové řešení Helix Broadcaster, tak i samotný streamingový server Helix Media Server a další doplňky v podobě Helix Media Library pro organizaci jednotlivých videí. Součástí aplikačního portfolia je také nepostradatelný Helix Producer, který se stará o překódování jednotlivých videí do požadovaných formátů. [29]

#### 4.1.1 Helix Server

Helix Media Server je zástupce streamovacího serveru pro doručení obsahu na různé zařízení od společnosti RealNetworks.

Helix RealMedia Server je základním produktem, který umí streamovat jak live, tak VoD obsah. Startovní edice je vykoupena mnohými omezeními, hlavními rozdíly oproti vyšší verzi je chybějící podpora pro:

- Flash, MPEG4, QuickTime, MP3
- Flash dynamic streaming (RTMP, RTMPS, RTMPT)
- Windows Media Streaming (Silverlight)

- 3GPP Format Delivery (On-Demand)
- Apple HTTP Live Streaming (Iphone, Ipad a další IOS zařízení)
- Encrypted Apple HTTP Live Streaming (HTTPS, AES)
- Network DVR (HLS time-shifting a H.264 server-side archiving)
- Apple HTTP Live Adaptive Streaming
- Fast Channel Switching (RTSP)
- Secure URL Authentication
- Client Session Control nad Logging
- Verimatrix DRM

Vyšší verze helix Universal Server se od nejvyšší Helix Universal Server Professional neliší v mnohém jiném, než v podpoře posledních tří výše uvedených vlastností.

Přehlednou tabulku můžeme vidět přímo na url:  
 „<http://www.realnetworks.com/uploadedFiles/Helix/Resource-Library/Datasheet-Helix-Server-Features.pdf> „

Helix Universal Server můžeme provozovat na těchto operačních systémech:

- Microsoft Windows Server 2008 R2 64-bit
- Red Hat Enterprise Linux Server v5-v6 64-bit
- Oracle Linux v6 64-bit
- Oracle Solaris 10 64-bit
- CentOS 5 64-bit
- CentOS 6 64-bit

Hardwarové požadavky jsou následující:

- 3,2 GHz Intel Pentim 4 Procesor (dual Xeon nebo výkonnější)
- SPARC Tx série nebo lepší
- 32-bit operační systém: 2GB RAM (4GB RAM)
- 64-bitový operační systém: 4GB RAM (8GB RAM)
- doporučena 1 GB síťová karta

Real Networks server využívala také Česká televize pro iVysílání spolu s dalším formátem Windows Media. [17]

## 4.1.2 Helix Broadcaster

RealNetworks nabízí mimo streamingový software i kompletní hardwarové řešení v podobě Helix Broadcasteru. Zde má zákazník na výběr ze 6 řešení (Series 100-600), kdy se cena toho nejlevnějšího pohybuje od necelých 5 tisíc dolarů a cena nejdražší verze je od 19 995 dolarů. Jednotlivé verze a specifikace jsou v přehledové tabulce dostupné na url: <http://www.realnetworks.com/helix/helix-broadcaster-editions.aspx>.

Tato řešení se liší maximální počtem streamových výstupů, kdy nejlevnější varianta nám umožní 2 HD 1080p (6Mbps) nebo 4 HD 720p (4Mbps) nebo 8 SD 480p (2Mbps) streamů, naopak nejvyšší konfigurace nám umožní až 100 HD 1080p (6Mbps) nebo 200 HD 720p (4Mbps) nebo 400 SD 480p (2Mbps) streamů. Samozřejmě můžeme použít i horší kvalitu streamů, kdy se server dostane mnohem více výstupů.

Dále se tyto verze také liší počtem vstupů, kdy jsou dostupné 1,2 nebo 4 vstupy s následujícími rozhraními:

*Video:* SDI, HD-SDI w/genlock, HDMI, DVB-C, Digital QAM (Annex A,B,C), DVBS-S2, DVB-T, DVB-ASI, 8VSB 9SMPTE 310) Analog (component/composite)

*Audio:* SDI Embedded, AC3 Embedded, AES/EBU

*IP Inputs:* UDP, RTP, RTSP, HTTP, HTTP Live (HLS), FTP, RTMP, Windows Media Video 9 with HTTP/ASF

Na výstupu máme na výběr z těchto možností:

*Kodek:* H.264, AAC, Ogg Vorbis, MPEG-1 LayerII, MP3 Dolby Digital AC3, H.263, VC1

*Formát:* MP4, FLV, F4V, 3GP, 3GP2, WMV, QuickTime, RealAudio, RealVideo, MP3, SMIL

*Protokol:* UDP, RTP, SDP, RTSP, HTTP, HTTP Live (HLS), HTTP Smooth Streaming, MPEG-DASH, RTMP, RTMPT, RTMPS, FTP, MMS, Windows Media Video 9 w/HTTP/ASF, IP-Unicast, IP-Multicast [30]

## 4.2 Windows Media Services

Microsoft přišel na trh s produkty serverové aplikace pro streaming Windows Media Services a enkódovacího softwaru Windows Media Encoder.

## 4.2.1 Windows Media Services

Vzhledem k velké rozšířenosti operačního systému Windows, ve kterém je také zabudován Windows Media Player se tento formát stal hojně využívaným. Nejznámější verzí serverové aplikace byla komponenta Windows Media Services, volitelně instalovatelná v rámci operačního systému Windows Server 2000, Windows Server 2003 a později také Windows 2008.

V dnešní době se již mnoho nevyužívá hlavně z důvodu nemožnosti adaptivního streamingu. Zde je uvedena zejména kvůli upozornění na platformu společnosti TV2Moro, která Windows Media services využívá hlavně z důvodu možnosti tzv. Time-shift nahrávání a následného „znovuspuštění“ o několik hodin později. Projekt TV2Moro je podrobněji představen v kapitole 8.2.

První verze Windows Media Services 4.1 byla součástí Windows Server 2000, ale s nástupem Windows Server 2003, přesněji v roce 2002 jej nahradila novější verze Windows Media Services 9 Series. Tato novější verze samozřejmě přinesla několik vylepšení, přičemž ty základní jsou:

- možnost instalace dalších rozšíření pomocí plug-inů
- zlepšení propustnosti I/O (Input-Output) a threading
- vylepšené uživatelské rozhraní poskytující rychlejší streaming skládající se ze čtyř částí:
  - rychlý start
  - rychlá mezipaměť
  - rychlá obnova připojení
- méně diskového vyhledávání v důsledku zvětšení datových bloků
- zvýšení počtu simultánních uživatelů
- in-memory caching nejčastěji přístupovaného obsahu díky použití vyrovnávací paměti operačního systému Windows Server 2003
- podpora pro Real Time Streaming Protocol (RTSP)
- lepší vytižení simulačního nástroje (Windows Media Load Simulator). [32]

S nástupem OS Windows Server 2008 přišla i nová verze Windows Media Services 2008, která ovšem stále nepodporuje možnost adaptivního streamingu a proto také nejsou Windows Media Services součástí Windows Server 2012.

Z hlediska rozdílnosti od předchozí verze Windows Media Services 2008 měla spíše zjednodušit natavení HTTP Sys konfigurace, a konfigurace Firewallu, kdy není potřeba

přidávat tento program do výjimek. Dále přibyla testovací utilita, zmenšila se latence spouštění streamů (Advance Fast Start) a služba byla přidána do QoS (Quality of Service) pro odchozí síťové rozhraní.

Windows media Services 2008 podporuje streaming těchto formátů:

- Windows media Audio (WMA)
- Windows Media Video (WMV)
- Advanced System Format (ASF)
- MP3
- Windows Media playlist (WSX). [33]

## 4.2.2 Windows Media Encoder

Windows media Encoder je software pro zachycení videa, nebo případně jen zvuku, z externího zdroje nebo ze souboru a jeho následné překódování do požadované kvality. Takto překódované video bývá většinou použito jako vstupní soubor nebo stream pro Windows Media Services.

Window Media Encoder 9 Series je podporován pouze na starších systémech Windows XP a Windows 2000, a proto je v dnešní době nahrazen novějšími verzemi popsanými níže.

V minimální konfiguraci, která potřebuje výpočetní výkon s procesorem o taktu 266 MHz a 64 MB RAM paměti, doporučená je však konfigurace s procesorem 500MHz a vyšším a alespoň 128 MB RAM paměti.

Windows media Encoder podporuje stejné formáty jako Windows Media Services – tj. Windows Media Audio (WMA), Windows Media Video (VMW), Advanced Format (ASF), MP3 a Windows Media playlist (WSX). [34]

## 4.3 Microsoft Smooth Streaming

Microsoft si metodu adaptivního streamingu pojmenoval Smooth streaming. Tato metoda je podporována skrze IIS Media Services a Expression Encoder.

### 4.3.1 Microsoft IIS Media Services

IIS Media Services je prvním produktem Microsoftu, který podporuje adaptivní streaming. Poprvé byl použit pro přenos letních olympijských her v roce 2008 pro portál NBCOlympics.com.

Ve Smooth Streaming klienti využívají jednoduchého konceptu doručení malých kousků videa (fragmentů), typicky se jedná o 2 sekundové části. Fragments jsou následně kontrolovány, zda dorazily ke klientovi včas a zda mohou být přehrány v požadované kvalitě. Pokud některý fragment nesplní tyto požadavky, další fragment bude zaslán v nižší kvalitě a následně, pokud to propustnost linky znovu dovolí, tak se kvalita sledovaného videa později opět přepne na vyšší.

IIS Media services však není založena na klasických streamovacích protokolech (RTSP, MMS, RTMP, apod.), ale na HTTP protokolu. Pro volbu HTTP protokolu hrály zejména tyto aspekty:

- HTTP (WEB) stahovatelné služby byly nabízeny poskytovateli CDN a hostingovými providery za nižší cenu
- streamovací protokoly měly často problém s prostupností přes firewally a routery, protože byly založeny především na UDP paketech s neobvyklými porty. HTTP používá klasický port 80, stejně jako ostatní webové stránky
- HTTP streaming nepotřebuje speciální proxy nebo cache servery
- Přenos HTTP dat na edge servery<sup>14</sup> je mnohem jednodušší a cenově výhodnější

Microsoft pro Smooth Streaming použil nový kontejner MP4 namísto staršího ASF z důvodu menší režie, snazšího analyzování a široké podpory třetích stran. MP4 je vyladěn pro podporu H.264, který se používá v širokém spektru Microsoft produktů včetně Silverlight, Windows 7, Xbox 360 Zune a Mediaroom.

Streamingové formáty Smooth jsou v podstatě dva: síťový formát (wire) a disk file formát. Při enkódování se video nahrává do jednoho dlouhého souboru na disk (jeden soubor pro každou kvalitu), ale ke klientovi je následně obsah doručován jako série malých chunků. Wire formát definuje strukturu jednotlivých chunků, které jsou posílány IIS klientovi, zatímco disk file formát definuje strukturu souvislého souboru na disku, což zlepšuje správu souboru. MP4 specifikace totiž dovoluje vnitřní organizaci série fragmentů což znamená že Smooth Streaming wire formát je přímou podmnožinou disk file formátu.

---

<sup>14</sup> Tedy blíže ke koncovému uživateli

Typické soubory potřebné ke Smooth Streamingu (Media Assets) jsou:

- MP4 soubory obsahující video/audio
  - \*.ismv - obsahuje video i audio nebo jen video, jeden soubor pro každou kvalitu
  - \*.isma – obsahuje pouze audio; v případě, že audio není samostatně, ale součástí videa, může být audio součástí ISMV souboru namísto separátního ISMA souboru
- server manifest file
  - \*.ism – popisuje vazby mezi mediem, přenosovou rychlostí a souborem na disku; je používán pouze IIS Smooth Straming serverem, není využíván klientem
- klient manifest file
  - \*.ismc – popisuje dostupný stream klientovi: použitý kodek, bitrate, rozlišení, makery (značky) apod., je doručen klientovi jako první soubor. Manifest soubry jsou XML formátované soubory, založené na SMIL 2.0 XML specifikaci. [35]

### 4.3.2 Expression Encoder

Důležitým prvkem pro Smooth streaming je také příprava obsahu, o něž se v případě smooth streamingu stará Expression Encoder, v dnešní době dostupný ve verzi 4.

První verze byla vydána v roce 2007, ale hned po roce byla dostupná verze druhá, která kromě jiných zlepšení přidala podporu H.264/AAC. [36]

Pozdější třetí verze byla vydána v roce 2009, a přinesla vylepšenou podporu H.264, byl přidán multikanálový audio výstup a podpora profilů pro streamování na zařízení jako Xbox, iPod, Zune a pro online služby Facebook a Youtube. [37]

Nejaktuálnější verze 4, která vyšla v roce 2010 a do roka se dočkala již druhého service packu. Expression Encoder nemá vysoké minimální nároky na instalaci - dá se provozovat na Windows 7, Windows Server 2008, Windows Vista, Windows XP Service Pack 3, přičemž další minimální požadavky jsou:

- počítač s procesorem alespoň 1GHz
- 1 GB Ram paměti
- 2 GB místa na disku
- .NET Framework 4.0

- Silverlight 4.0
- podpora Microsoft DirectX 9.0 grafiky s Windows Vista Display Driver modelem (WDDM) a 128 MB RAM paměť
- DVD kompatibilní mechanika
- monitor s rozlišením 1024 x 768 ve 24 bitovém barevném podání
- internetové funkce jsou dostupné pouze s přístupem na internet
- některé vlastnosti prvků potřebují FireFox 3.0 a vyšší nebo internet Explorer 8.

Expression Encoder 4 má na výběr několik vstupních formátů videa jako: .asf, .avi, .m2v, .mts/.mts2 (AVCHD), .wmv, wtv (Windows Television), .xesc (Expression Encoder screen capture format), přičemž další formáty .3g2, .3gp, .avs, .dv, .dvr-ms, .m4v, .mod, .mov, .mp4, .mpeg, .mpg, ts2, vob podporuje s pomocí Apple Quicktime, MPEG-2 dekodérem nebo AviSynth. Z obrazových formátů podporuje snad všechny základní a co se týče audio formátů, tak na výběr z: .ac3, .aiff, .m4A, .m4B, .bwf, .mp3, .mp4, .wav a .wma. [38]

## 4.4 Quick Time Streaming Server

Quick Time od společnosti Apple je další alternativou jak streamovat video. Společnost Apple vyvinula Quick Time Streaming server pro MAC OS X Server. Pomocí Quick Time s příponou \*.mov podporuje standard 3GPP a samozřejmě je také MPEG-4 komprese.

Tento streaming server se v dnešní době již zřejmě nepoužívá, protože lze nainstalovat pouze na operační systém MAC OS X Server verze 10.6 Snow Leopard z roku 2009, přičemž nyní nejnovější verze, která vyšla v říjnu 2013, je již MAC OS X Server 10.9. Vzhledem k této skutečnosti se produktem zde dále zabývat nebudeme a raději se zaměříme na jiný streaming SW z portfolia Applu. [22] [39]

## 4.5 Quick Time Broadcaster

Quick Time Broadcaster je spíše rozšíření pro Quick Time pro domácí použití, nejedná se o žádnou sofistikovanou aplikaci. Má velké omezení v podobě vstupu, protože lze použít pouze iSight kameru<sup>15</sup> nebo případně vstup DV pomocí Fire-Wire portu. Quick Time

---

<sup>15</sup> webkamera na počítačích od společnosti APPLE



je volně dostupný a lze jej doinstalovat na verze MAC OS X 10.4 a pozdější. Nejaktuálnější verze 1.5.3 podporuje mimo jiné také H.264 streaming a má podporu 3GPP. [22], [39]

## 4.6 HTTP Live Streaming a Darwin Streaming Server

Společnost Apple má také profesionální produkty pro streamování a to v podobě aplikací HTTP Live Streamingu nebo Darwin Streaming Serveru. Tyto dva produkty jsou hodně podobné, pouze v případě Darwin Streamingového Serveru se jedná o open source.

Darwin Streaming Server je vhodné řešení pro streaming formátu QuickTime a MPEG-4 pomocí protokolu RTP/RTSP pro alternativní operační systémy jako Windows, Linux nebo Solaris.

HTTP Live Streaming (HLS) je založen na klasickém HTTP Serveru, který doručuje již upravený obsah. V případě live streamu je vstup brán pomocí Media Stream Segmenter z MPEG-2 transport streamu a následně je tento obsah rozdělen na sérii stejně dlouho trvajících časových úseků potřebných pro HTTP live streaming.

Media Stream Segmenter může také vygenerovat index file (playlist), zakódovat data a vygenerovat dekodovací klíč, potřebný k přehrání.

Oproti tomu existuje Media File Segmenter, který použije jako vstup již vytvořený videosoubor, zabalí ho do MPEG-2 transport streamu a poté jej rozdělí na segmenty. Pro validaci odkazů nám slouží Media Stream Validator, pro vygenerování playlistu Variant Playlist Creator a pro vkládání metadat zase Metadata Tag Generator.

HTTP Live streaming se využívá hlavně pro streaming pro zařízení s iOS systémem, viz níže uvedené výrobcem definované profily pro jednotlivé typy zařízení.

Video stream využívající H.264 kompresi:

- H.264 Baseline 3.0 : Všechna zařízení
- H.264 Baseline 3.1: iPhone 3G a novější, iPod touch 2 generace a novější
- H.264 Main profile 3.1: iPad (všechny verze), Apple Tv 2 a novější, iPhone 4 a novější
- H.264 Main Profile 4.0: Apple TV3 a novější, iPad 2 a novější, iPhone 4S a novější
- H.264 High Profile 4.0: Apple TV3 a novější, iPad 2 a novější, iPhone 4S a novější
- H.264 High Profile 4.1: iPad 2 a novější, iPhone 4S a novější

Doporučený frame rate<sup>16</sup> je 10 fps pro datový tok menší než 200 kbps, 12-15 fps pro stream menší než 300kbps a 29,97 fps pro všechny ostatní.

Audio stream může využít jeden z následujících formátů:

- HE-AAC nebo AAC-LC, stereo
- MP3 (MPEG-1 Audio Layer 3), stereo [22], [39], [40]

## 4.7 Adobe Flash Video

Společnost Adobe (dříve Macromedia) začala konkurovat společnosti Microsoft v oblasti streamingu. Postupně si získala oblibu na trhu a nyní je zřejmě nejpoužívanější platformou pro streaming videí. Adobe Flash video s příponou \*.flv si umí poradit s mnoha kompresemi včetně MPEG-4.

Adobe nabízí pro streaming Adobe Media Server 5, jehož minimální požadavky na výkon hardwaru jsou:

- 3,2 GHz Intel Pentium 4 procesor (dual Intel Xeon nebo rychlejší)
- Microsoft Windows Server 2008 R2 (64bit), Red Hat Enterprise Linux Server 5.5 (64bit) nebo Linux CentOS 5.8 (64bit)
- 64-bitový operační systém se 4 GB RAM pamětí (doporučeno 8GB a více)
- 1GB síťová karta (konfigurace s více kartami i 10 GB síťová karta je také podporována)

Media Server má více variant, odlišujících se funkcemi – přehledně jsou tyto funkce zobrazeny na webu, případně v manuálu Adobe.

Nejvíce limitovaná verze je Adobe Media Server 5 Starter, která je spíše pro testování, protože je omezena délkou videa v případě HLS na 10 minut, verze Standard pak již omezena délkou videa není.

Nejvíce funkcí nabízí verze Extended a o něco málo méně verze Professional. Tato licence stojí přibližně 4500 dolarů.

Adobe nám nabízí více možností jak doručovat video ke koncovému klientovi, přičemž každá má své výhody:

- RTMP Dynamic Streaming (Unicast) – vysoce kvalitní, nízká latence streamingu s podporou live streamingu a VoD a plnou podporou adaptivního bitrate
- RTMPE (Encrypted RTMP) – real-time šifrované RTMP

---

<sup>16</sup> počet snímků za sekundu

- RTMFP (multicast) – IP multicast šifrování s podporou pro ASM i SSM multicast pro multicastové sítě
- RTMFP (P2P) – P2P live video stream mezi dvěma Flash přehrávači (klienty)
- RTMFP (multicast fusion) – IP a P2P spolupracující navzájem pro lepší QoS v rámci sítě
- HTTP Dynamic Streaming (HDS) – povolení VoD a live adaptivního bitrate streamingu založeném na standardu MP4
- Protected HTTP Dynamic streaming (PHDS) – real-time šifrování HDS
- HTTP Live Streaming (HLS) – HTTP live streaming pro iOS zařízení nebo zařízení která podporují formát HLS; volitelné šifrování pomocí AES128

Adobe ke zpracování videa používá kodeky jako Sorenson Spark, On2 VP6-S, On2 VP6-E nebo H.264 pro video a Nellymoser, MP3, AAC nebo Speex pro audio. [20],[41]

## 4.8 Wowza

Wowza je produktem firmy s Wowza Media Systems LLC, která byla založena v roce 2005 ve státě Colorado, USA. Již v roce 2007 vydala první verzi Wowza Media serveru jako levnější alternativu k Adobe Flash Media Serveru. V roce 2009 představila již druhou verzi s podtitulem „H.264 everywhere“, tedy H.264 všude.

Wowza 2.0 se rozšířila díky podpoře Apple HTTP Live streaming protokolu pro zařízení iOS, Microsoft HTTP Smooth streamingu pro Silverlight přehrávače, RTSP/RTP pro QuickTime přehrávače, 3GPP podpoře pro mobilní zařízení s Android nebo BlackBerry (RIM) systémem a také díky podpoře IPTV/OTT set-top boxů a herních konzol.

V roce 2011 Wowza představila již třetí verzi, která nabízí jednotný rozšířitelný základ s možností dokoupení funkcí pro doručení obsahu do všech možných zařízení. V této třetí verzi je již implementována možnost adaptivního bitrate (ABR), time-shift přehrávání a jednoduchá a efektivní ochrana digitálního obsahu.

Nyní je na trhu poslední verze s označením 3.6, která se pyšní používáním více než 150 000 licencí ve více než 150 zemích světa. Hlavním rozdílem od předchozí verze je podpora dynamického adaptivního streamingu pomocí HTTP (DASH).

Na rozdíl od dalších dvou nejpoužívanějších platforem Adobe Media Server a Microsoft IIS Media Services podporuje Wowza podporuje Quicktime/3GPP (RTSP,RTP), IPTV (MPEG-TS), MPEG-DASH a je zejména multiplatformní (podporované operační systémy jsou Windows, Linux, Solaris, Unix i Mac OS X).

Minimální požadavky pro instalaci jsou: procesor Quad Core 3 GHz a vyšší, dva a více disků v RAID 0, síťová karta 1Gb.

Wowza podporuje streaming Adobe Flash RTMP (RTMPE, RTMPT, RTMPTE, RTMPS), Adobe HTTP Streaming (HDS) dále také Apple HTTP Live Streaming (HLS), MPEG-DASH, RTSP/RTP, Microsoft Smooth Streaming i MPEG2 Transport Protocol (MPEG-TS).

Cena Wowza licence začíná na 5 dolarech za den, případně měsíční licence vyjde na 55 dolarů anebo se můžeme rozhodnout pro koupi časově neomezené licence za 995 dolarů.

Mezi zásuvné moduly Wowzy patří:

- Transcoder overlays – tento modul umožňuje vložení reklamy, vodoznaku, loga nebo symbolu
- StreamLock – modul umožňující zabezpečení pro síťové šifrování pomocí 256 bitového SSL certifikátu
- DRM – modul umožňující integraci šifrování pomocí DRM a buyDRM [42], [43]

## 4.9 VideoLan VLS a VLC

Videolan začal jako školní projekt na vysoké škole École Centrale Paris ve Francii v roce 1996, po jeho kompletním přepsání a uzavření dohody se školou se z něj stal v roce 2001 open source projekt.

Projekt se otevřel pro vývojáře mimo École a nyní je z něj již celosvětový projekt s vývojáři z více než 20 zemí světa. Od roku 2009 je Videolan kompletně oddělen od École Centrale Paris a stala se z něj autonomní nezisková organizace.

VideoLan přišel nejdříve se svým VLS (VideoLan Serverem), který byl pod licencí GNU-GPL. Jeho výhodou bylo, že byl multiplatformní a uměl pracovat s řadou kompresních formátů včetně Ogg Theora. VLS však již není dále vyvíjen, protože jej nahradil multiplatformní přehrávač VLC s možností streamingu.

Tento multimediální přehrávač má podporu snad pro všechny známé systémy (Windows, Mac OS X, různé verze Linuxu i ostatní systémy jako Solaris, Andorid, iOS atd.), což následně umožňuje streamování z jakéhokoliv podporovaného systému.

VLC podporuje protokoly jako UDP unicast/multicast, RTP Unicast/multicast, HTTP i MMS. Ze streamovacích kodeků máme na výběr z:

- MPEG-1, MPEG-2, MPEG-4

- DivX 1, DivX 2, DivX 3
- H.263, H.264
- WMV1, WMV2
- MJPEG
- Theora
- Dummy

VLC na poli komerční sféry nepředstavuje velkou konkurenci k ostatním aplikacím, i když je pravda, že je to zajímavá varianta minimálně z toho důvodu že je open source, tedy zdarma. [21]

## 5 Streamovací řetězec

Tato kapitola se zabývá kompletním řetězcem pro streaming.

Základem streamingu je zdroj videa, které server zpracuje nejčastěji za pomoci karty s externím vstupem pro získání obsahu například z DVD, satelitního či televizního vysílání nebo přímo z digitálního souboru.

Server si můžeme představit jako vrchol pyramidy, pod kterým obvykle bývá více distribučních serverů, které dohromady nazýváme CDN<sup>17</sup>. Distribuční servery pak „přebírají“ obsah z primárního serveru a posílají je dále k divákovi. Tento princip se hojně využívá hlavně v případech, kdy je o obsah daného streamingu velký zájem, a jeden server by nezvládl velkou zátěž, pohybující se například v desítkách tisíců diváků.

Datový tok streamingu může být odolný vůči ztrátám přenosu (MPEG-PS, RTSP, RTCP), kdy fungují mechanismy pro kontrolu paketů nebo naopak v případech, kdy pakety nemusí být chráněny (RTP, MPEG-PS).

V zásadě jde o to, zda se provádí nějaká kontrola přenesených paketů, zda je transport videa orientován pouze na jednoho či více lidí (unicast, multicast), s čímž zase souvisí rozdíly mezi UDP a TCP. Podrobněji se tímto tématem zabývá kapitola 5.2.

### 5.1 Zdroj videa

Zdroj videa či audia je základem každého streamingu. Většinou na úplném základu bývá videokamera, ale nemusí tomu tak být vždy – například při animované tvorbě. Pro účely této práce bude zdrojem videa jednoduše obsah, který se má streamovat, nezávisle na konkrétním způsobu vytvoření či získání onoho obsahu.

Z hlediska streamování je potřeba rozlišit pouze to, zda je zdroj videa k dispozici ve své kompletní podobě (například soubor uložený na disku), či zda je k dispozici jen jeho část – streamuje se např. z DVB-T vysílání.

První případ je nejjednodušší, protože většina streamovacích aplikací si umí poradit s digitálními soubory nejpoužívanějších formátů a kompresí, případně existují softwary, které tyto videa umí převést do formátu, který je kompatibilní se streamovacím softwarem.

Potíž může nastat v případě, kdy se jedná o streaming z digitálního pozemního nebo satelitního vysílání, případně při použití kamer nebo zařízení využívající profesionální výstupy, jako například SDI. Videosignály vysílané tímto způsobem musíme nejdříve

---

<sup>17</sup> Content Delivery Network

nějakým způsobem převést do počítače, mnohdy za použití speciálních přídatných karet přizpůsobených těmto účelům.

### 5.1.1 Signály DVB-T/2, DVB-S/2, DVB-C

Pro příjem z digitálního vysílání potřebujeme nejdříve zařízení, které nám tento příjem umožní tj. příslušnou anténu a zařízení, které umí přijímaný signál zpracovat. Samozřejmě lze z cenových důvodů zvolit neprofesionální přijímače, které jsou určeny pro domácnosti, ale tím se ochudíme většinou o možnosti například vzdáleného ovládání pomocí webového rozhraní – což při větším počtu kanálů může být problém. Nehledě na to, že tato zařízení nemusíme mít přímo v dosahu, ale mohou od nás být vzdálené několik set nebo i tisíc kilometrů.

Proto se v praxi využívají profesionální IRD<sup>18</sup>. Na výběr máme z několika možností. Mezi nejpoužívanějšími patří receivery od firem jako Ericson<sup>19</sup>, Harmonic, Teracue a pak dále například receivery od firem Scopus, Advanced Digital, 2Wcom, Wellav Technologies, Upcom Technologies a mnoho dalších více či méně známých firem, které se poslední dobou snaží dostat na výslunní anebo alespoň do podvědomí broadcasterových firem.

Nejnovější receivery mají na výběr obvykle z několika výstupů, mezi které z velké části patří digitální SD/HD/3G-SDI s embedovaným audiem, kompozitní videovýstup, případně ještě další výstupy jako například MPEG-TS, který lze využít přímo jako vstup pro streamovací aplikaci.

### 5.1.2 Hardwarové karty pro přijímání signálů

Zařízeními s předchozí kapitoly získáme videosignál, který potřebujeme nějakým způsobem dostat do počítačového softwaru.

V případě že IRD disponuje digitálním výstupem MPEG-TS nepotřebujeme již žádný další hardware. Mnohdy ale můžeme potřebovat dostat signál z receiveru, kamery, střížny či dalšího zařízení disponující pouze SDI, kompozitním, komponentním nebo jiným do počítače. Mezi známé výrobce takovýchto vstupních karet disponujícími celou škálou videovstupů patří karty DeckLink od firmy BlackMagicdesign, Osprey od společnosti ViewCast, Bluefish od společnosti Bluefish444, VisionAV-SDI od Datapath, Quadro SDI Capture od výrobce grafických karet Nvidia a další.

---

<sup>18</sup> Intergrated receiver/decoder

<sup>19</sup> Firma Ericsson odkoupila firmu Tandberg, která vyráběla stejnojmenné receivery dříve

Tyto přídatné karty se instalují do počítače nebo serveru pomocí rozhraní PCIe. Při výběru konkrétního výrobce a typu je důležité zjištění kompatibility mezi danou kartou a softwarem, ve kterém jí budeme používat.

## 5.2 Úprava a kódování videa

V případě že již máme vstupní zdroj, případně soubor k dispozici, většinou jej potřebujeme nějakým způsobem upravit. Úpravou můžeme rozumně například snížení bitového toku, překódování do správného kodeku, změnu rozlišení, přidání loga či nějaké grafiky případně přípravu zdroje pro adaptivní streaming.

Ve většině streamovacích softwarů máme tyto jednotlivé možnosti v nabídce, tudíž stačí pouze správně nastavit parametry a na výstupu již budeme mít hotový stream v námi požadovaném datovém toku a rozlišení. Pokud však nechceme využívat streamovací software, máme možnost předpřipravít si vstupní zdroj nějakou aplikací, například pomocí programu ffmpeg.

### 5.2.1 FFmpeg

FFmpeg vznikl původně pod GNU/Linuxem, ale nyní je podporován nejenom pod operačním systémem Linux, ale také pod operačními systémy Windows a Mac OS X.

Tento software je snad nejuniverzálnějším dekodovacím, kódovacím, transkodovacím a přehrávacím nástrojem, protože si dokáže poradit s většinou videokompresních formátů.

FFmpeg se snaží poskytovat nejlepší technicky možné řešení pro vývojáře aplikací a koncových uživatelů přičemž za něj nemusíme platit, protože je zdarma.

Tento program se ovládá pomocí příkazové řádky, přičemž si kdokoliv může naprogramovat své grafické rozhraní, případně použít nějaké z volně dostupných, kde však nemusíme mít zaručenou jak přesnost nastavení tak i bezpečnost. Podrobný manuál programu, kde jsou vysvětleny jednotlivé přepínače i příkazy je dostupný na oficiálních stránkách FFmpeg.

Podrobněji se mu budeme věnovat dále v praktické části, kde si názorně předvedeme jednotlivé příkazy pro transkodování videa do námi určeného nastavení. [44]



## 5.3 Streamingový server - Origin

Origin je název pro server (skupinu serverů), který má přístup k originálnímu zdroji videa, ať už v podobě již připraveného streamu nebo souboru. Tento server pomocí streamovacího softwaru případně pomocí obyčejného web serveru streamuje jednotlivá videa.

V robustnější struktuře, kdy počítáme s větším zájmem diváků však tento server poskytuje data pro další servery a ne přímo pro koncové klienty. Tímto se přechází možnému výpadku serveru z důvodu přetížení. Počet doručovacích CDN serverů bývá v závislosti na předpokládaném počtu jednotlivých streamů, které si klienti pustí v jeden okamžik.

## 5.4 CDN

Jak je již zmíněno v přechozí kapitole počet CDN serverů může být různý – jiný počet bude mít regionální vysílání oproti vysílání olympiády či jiných mezinárodně sledovaných událostí.

Vlastnictví takové sítě, byť i jen několika desítek serverů, není levnou záležitostí. CDN servery navíc z důvodu konektivity obvykle nebývají umístěny doma, či přímo ve firemním prostředí, ale ve větších či menších datových centrech na páteřních linkách internetu.

Za pronájem serveru spolu s pronájemem dostatečně širokého datového pásma firmy platí nemalé peníze – samozřejmě v závislosti na cenách jednotlivých poskytovatelů.

Mimo fyzické vlastnictví CDN je zde možnost pronájmu již existujícího CDN, případně nějaké jeho „virtuální“ podoby. Odpadne tak mnoho starostí jak po stránce hardwarové, tak i po stránce softwarové.<sup>20</sup>

Tyto virtuální, neboli cloudové, CDN mohou pomoci nejen s distribucí obsahu k většímu počtu uživatelů, ale mohou také hravě nabídnout výpočetní výkon potřebný k překódování videa do jiných kvalit či formátů. Využití záleží pouze na konkrétních požadavcích a potřebách.

Mezi nejvýznamnější CDN providery videa bezesporu patří:

Akamai, EdgeCast Networks, Highwinds, Limelight, Mirror Image, iStreamPlanet, Octoshape, PowerStream, StreamGuys, Streamzilla, CDN77, Level3 a spousta dalších.

---

<sup>20</sup> Například instalace updatů a jiných bezpečnostních mechanismů

V případě, že se jedná o streaming pro mobilní zařízení, vstupuje do řetězce ještě síť mobilního operátora, která zprostředkovává videosignál pomocí své sítě do mobilních zařízení.

## 5.5 Koncový uživatel

Dalším a již posledním článkem ve streamovacím řetězci je koncový uživatel k němuž video putuje z nejdostupnějšího CDN serveru.<sup>21</sup>

Koncový uživatel se na video může dívat prostřednictvím různých zařízení – ať už se jedná o různé mobilní telefony, tablety, herní konzole, počítače, či upravené set-top-boxy pro příjem streamů.

Může se jednat jak o bezplatnou tak i placenou službu a každá společnost má svou cenovou politiku stanovenou jinak.

V případě placeného obsahu mají společnosti vytvořenou databázi uživatelů nebo přístrojů vůči níž se ověřují přístupy k jednotlivým obsahům. Pokud tedy zákazník za placenou službu nezaplatil bývá odmítnut a přesměrován na odkazy, vysvětlující jak je možné si službu zakoupit.

Pomineme-li tedy fakt, že za streaming videa v některých případech musíme platit, koncovému uživateli stačí vlastnit kompatibilní přijímač a dostatečně rychlé internetové připojení – v závislosti na velikosti datového toku videa.

---

<sup>21</sup> Což nemusí být vždy nejbližší server! Záleží na routování sítí, peeringu atd.

## 6 Typické artefakty

### 6.1 Kompresní artefakty

Většina kompresních algoritmů používaných různými standardy je dost podobná. Mnoho z nich spoléhá na DCT s následným kvantováním koeficientů. V těchto kódovacích systémech jsou kompresní deformace způsobené pouze jedinou operací a to kvantováním koeficientů transformace. Ačkoliv další faktory ovlivňují výslednou kvalitu streamu (například pohybová predikce nebo velikost vyrovnávací paměti) nevnese narušení obrazu přímo, ale mají vliv na komprimaci nepřímo. [1], [2]

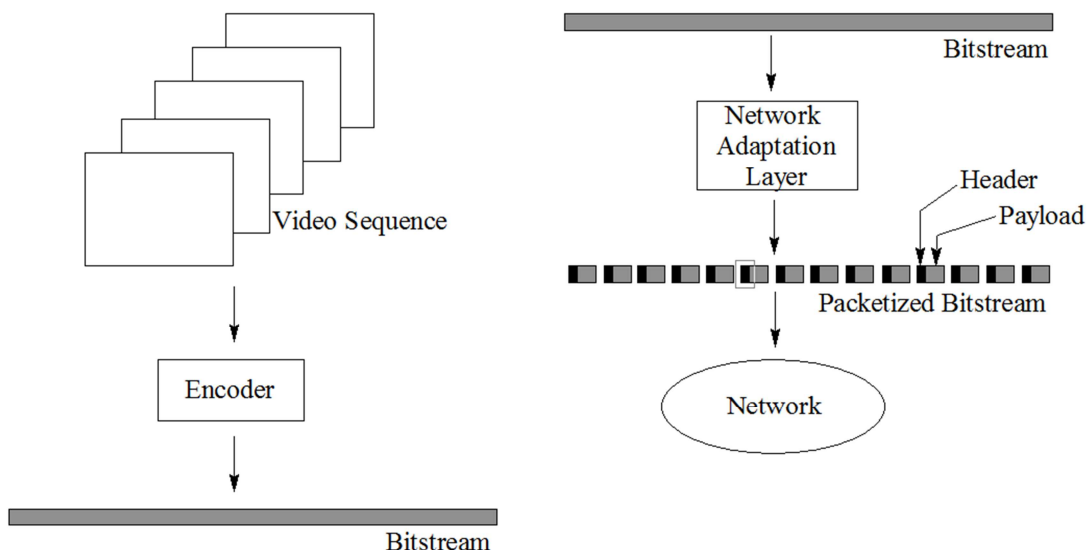
V komprimovaném videu můžeme rozlišovat tyto artefakty:

- Blokový efekt (čtverečkování) - je způsobeno nezávislým kvantováním jednotlivých bloků (obvykle 8x8 pixelů) v DCT kódovacím schématu, což vede k ostrým hranám na hranicích jednotlivých bloků. Tento efekt je často nejvýraznějším zkreslujícím artefaktem v komprimovaném videu vzhledem k rozsahu a pravidelnosti. Kodek H.264 se tento jev snaží potlačit deblokovacím filtrem.
- Rozostření (Blur) – vyznačuje se ztrátou detailů a snížením ostroty hran. To je výsledek potlačení vysokofrekvenčních koeficientů hrubým kvantováním.
- Color Bleeding - „roztírání“ barev mezi silně barevnými oblastmi. Toto je následek potlačení vysokofrekvenčního koeficientu barevné složky. Vzhledem k barevnému podvzorkování se tento jev projevuje na celém makrobloku.
- Slanted lines – „schodišťové linky“ jsou dány skutečností, že DCT je nejvhodnější na reprezentaci horizontální a svislé čáry, zatímco kterékoliv jiné směry vyžadují vyšší frekvence DCT koeficientů pro přesnou rekonstrukci. Typicky jsou proto šikmé čáry „zubaté“.
- Ringing – je spojen s Gibbsovým fenoménem a je tedy nejvíce patrný na vysoce kontrastních hranách v jinak hladké ploše. Vyskytuje se jak u jasové, tak i u barevné složky.
- Jagged motion – vyskytuje se díky špatnému odhadu pohybu. Tento odhad funguje dobře pouze tehdy, když jsou všechny pixely v makrobloku identické.

- Chrominance mismatch – je důsledkem metody odhadu pohybu, který je často prováděn na základě jasové složky.
- Mosquito noise – vyskytuje se často především na rozhraní hladkých struktur a hran s vysokým kontrastem nebo pohybujících se objektů.
- Flickering (blikání) se často vyskytuje při vysokém kontrastu struktury v obraze. Tento jev je důsledkem komprimace, kdy jsou různé bloky komprimovány a kvantovány s různými kvantovacími koeficienty.
- Aliasing - vzniká při nesplnění Nyquistovy podmínky.

## 6.2 Chyby způsobené přenosem

Důležitým, a často přehlíženým zdrojem poruch bývá zpravidla samotný datový stream. Digitálně komprimované video je obvykle přenášeno prostřednictvím paketových sítí. O jeho fyzickou přepravu, ať už se jedná o kabelovou (drátovou) nebo bezdrátovou se starají transportní protokoly jako například ATM nebo TCP/IP. Datový stream je přenášen po malých kouscích (paketech), které obsahují tzv. hlavičky (headery), jenž v sobě nesou informaci sekvenční a časovou. Toto je znázorněno na obrázku 5.1. [1], [2]



Obrázek 6.1: Příklad přenosové cesty videa při streamingu [2]

Streamy mohou nést další signalizační informace na úrovni relační vrstvy. Většina aplikací pracujících se streamingovým obsahem potřebuje možnost dekódovat stream v reálném čase, jakmile tyto pakety dojdou. Během cesty internetem (nebo lokální sítí) však mohou nastat dva problémy: některé pakety se mohou ztratit nebo dorazit ke koncovému zařízení se zpožděním. To je následkem řazení datových paketů v síťových zařízeních, jako jsou například routery nebo switche. Pro samotný dekodér a přehrávač to má stejný dopad, jako by paket s požadovanou informací nedorazil vůbec, protože jsou potřeba ve chvíli, kdy se mají dekódovat. Tímto mohou vznikat v obraze různé artefakty chybějících částí obrazu.

## 6.3 Ostatní poruchy

Mimo dvou dříve uvedených typů poruch může být kvalita videa ovlivněna také před nebo po kompresním procesu:

- konverzí mezi digitální a analogovou částí
- barevným podvzorkováním
- snímkovou konverzí mezi různými formáty zobrazení
- deinterlacingem – procesem, který z půlsnímkových videí vytváří celosnímkové (neprokládané). Takovým příkladem může být konverze mezi 24-snímkovým videem na 60-snímkové video.[1], [2]

## 7 Hodnocení kvality videa

### 7.1 Subjektivní hodnocení

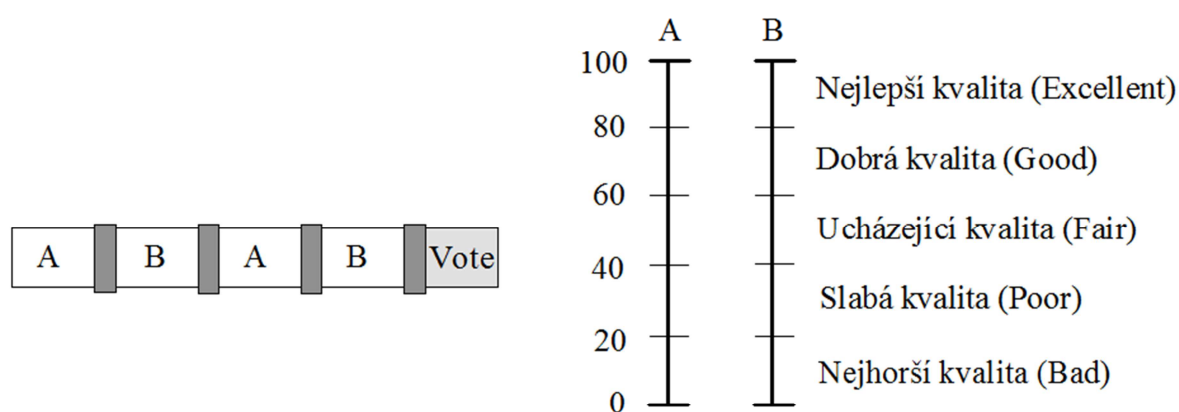
Subjektivní hodnocení kvality videa je asi nejúčinnější metodou hodnocení kvality je ale bohužel spjato s časovou náročností a také náročností na lidské zdroje, vzhledem k tomu, že výsledné video hodnotí lidé, pro které je video určené a ne algoritmy, kterým je v zásadě jedno jakou kvalitu bude přehrávané video mít.

Pro subjektivní hodnocení existuje několik metod, které podle doporučení ITU-R BT 500-11 popisují podmínky sledování, počet pozorovatelů, vyhodnocovací podmínky apod.

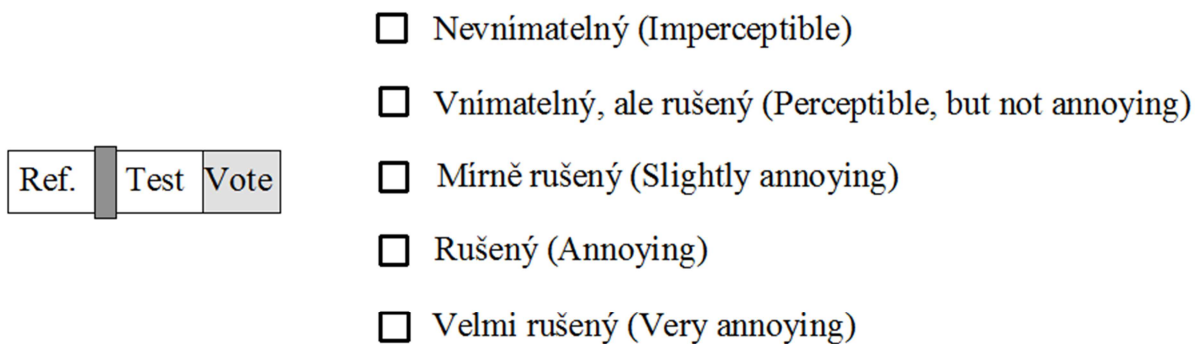
Kompletní popis všech metod subjektivního hodnocení by byl obsáhlý, a proto jsou zde uváděny pouze ty nejčastěji používané jako zástupci.

DSCQS (Double Stimulus Continuous Quality Scale) je metoda, kdy se divákům zobrazují postupně snímky originálu a testovaný snímek, mezi nimiž je proloženo video šedé scény po dobu 3 sekund.

DSIS (Double Stimulus Impairment Scale) metoda je oproti předchozí metodě jiná v tom, že sekvence originál-šedá-testované video se zobrazí pouze jednou a poté následuje hodnocení kvality. Obě dvě tyto metody můžete vidět na obrázcích 6.2 a 6.2 spolu se stupnicí hodnocení.



Obrázek 7.1: Příklad DSCQS metody (vlevo sekvence obrazů, vpravo hodnocení kvality) [2]



Obrázek 7.2: DSIS metoda (vlevo sekvence obrazů, vpravo hodnocení kvality) [2]

## 7.2 Objektivní hodnocení

### 7.2.1 MSE a PSNR

MSE (Mean Square Error) a PSNR (Peak Signal-to-Noise Ratio) jsou nejjednodušší metodou pro hodnocení kvality videa a jsou definovány následovně:

$$MSE = \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2$$

$$PSNR = 10 \log_{10} \frac{L^2}{MSE} \quad [1]$$

kde  $N$  je počet pixelů a  $x_i$  spolu s  $y_i$  jsou  $i$ -té pixely. Přitom  $x$  zastupuje pixely původního a  $y$  zkresleného (zkomprimovaného) obrázku.  $L$  je dynamický rozsah.

Tyto dvě metriky mají jasný fyzikální význam a z matematického hlediska se snadno optimalizují, i když nemusí úplně korespondovat s lidským vnímáním. [1], [2], [23]

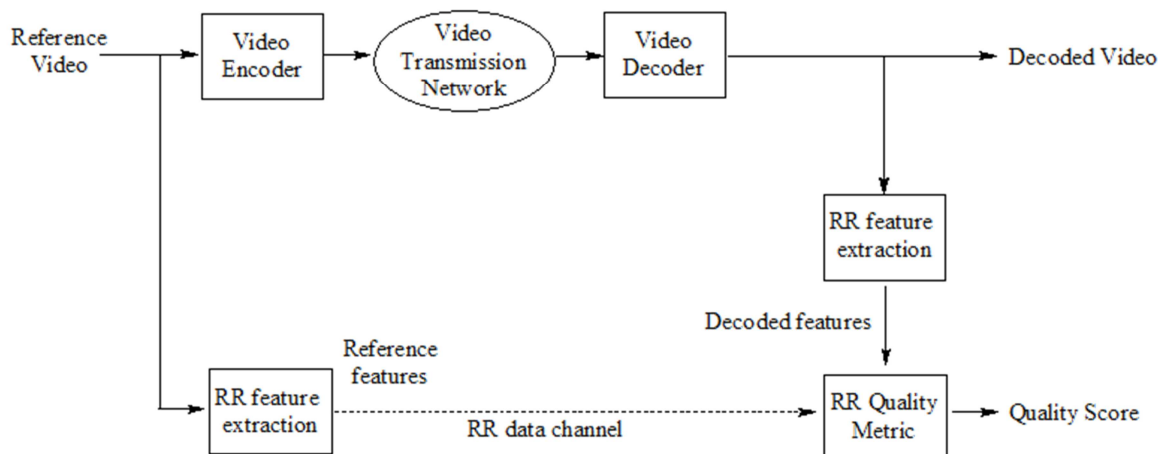
### 7.2.2 Měření kvality pomocí strukturálního zkreslení

Tento přístup pracuje na základě toho, že obraz z roztaženým kontrastem vnímáme jako více kvalitní než komprimovaný obraz s nižším kontrastem. Toto měření předpokládá, že HVS pracuje s predikcí vizuálního vnímání. [1], [2], [23]

### 7.2.3 Měření s redukovanou referencí

Předchozí měření kvalit bylo založeno na základě porovnávání komprimovaného a originálního (referenčního) videa.

Tato metoda předpokládá, že spolu se streamem videa jsou přenášeny také pomocné informace, které zároveň fungují jako referenční hodnoty pro hodnocení kvality videa. Na obrázku 7.3 je ukázáno, jak lze tuto metodu aplikovat.



Obrázek 7.3: Schéma pro metodu měření s redukovanou referencí [47]

### 7.2.4 Měření bez reference

Navrhování algoritmů pro hodnocení touto metodou je nejobtížnější, zejména z důvodu malého poznání HVS a příslušných mozkových vyhodnocovacích procesů.

Největším zdrojem zkreslení videa jsou blokové chyby způsobené DCT, které se snaží většina bezreferenčních algoritmů vyhledávat. Tyto artefakty jsou uvedeny v kapitole 6.1 – Kompresní artefakty.



## 8 Streaming v praxi

Pro využití streamingu v praxi byly zvoleny projekty, ke kterým byly získány informace nejenom v průběhu diplomové práce

Je logické, že si firmy střeží své know-how před konkurencí a že v důsledku těchto omezení není možno zveřejnit všechny získané informace.

### 8.1 iVysílání české televize

Tento projekt je v ČR asi nejstarším a nejdéle trvajícím na poli streamingu. Česká televize (ČT) spustila své internetové vysílání v roce 2005.

Původně začala ČT vysílat ve formátech Real Media a Windows Media ve dvou různých kvalitách videa. Postupem času se však situace vyvíjela a nyní se již používá Wowza server. Na webových stránkách ČT tak mohou diváci najít pořady, pro které má televize práva ke zveřejnění na internetu.

U nových pořadů, vysílaných ve flashi, je k dispozici video ve formátech: 720p (1280x720, 3500kbps), 576p (1024x576, 2000 kbps), 404p (720x404, 1000 kbps) a 288p (512x288, 500 kbps), pro mobilní zařízení pak 288p (512x288 500 kbps), 144 (256x144, 200kbps) a 144 (256x144, 100kbps).

K těmto video streamům je potřeba ještě připočítat stereo audio složku s datovým tokem 96kbps, případně u mobilních streamů mono 32kbps.

Video obsah je do streamingového řetězce nabírán buď přímo ze souborů z odbavovacího serveru ČT (v případě předtočených pořadů) nebo, v případě vysílání live, pomocí Osprey karty a signálu HD SDI přímo v sídle ČT.

Tato videa jsou následně enkódována do požadované kvality pomocí softwaru Flash Media Live Encoder (FMLE), případně pomocí FFMpegu, který je ovšem výpočetně náročnější.

Co se HW nároků týče, tak server s parametry CPU Intel Xeon x5650 (12 jader), 12 GB RAM dokáže najednou kódovat tyto kvality: 2x720p, 1x576, 2x404, 2x288.

Z hlediska technologií, které jsou pro iVysílání použity, to jsou zejména RTMP pro webové rozhraní, RTSP pro zařízení se systémem android a HLS pro Apple zařízení, doplněno progresivním downloadem pro televizní přijímače.

Se stejně vybaveným serverem, jako je server enkódovací, může Wowza server běžně odbavit až několik tisíc zákazníků. Wowza může na takovémto HW v průměru dosáhnout odchozí datový tok až okolo 3 GB/s.

K zajištění technického provozu streamingu ČT využívá externí firmu, která má svou vlastní CDN platformu čítající několik desítek serverů.

## 8.2 TV2Moro

TV2Moro projekt byl postaven na stejné platformě, tedy na Window Media Serveru, na kterém z části běží dodnes. Jedná se o zprostředkování televizního vysílání z Blízkého východu pro severoamerický kontinent.

Satelitní vysílání DVBS a DVBS-2 je pomocí parabol získáván jak v RRSatu v Izraeli, tak v ECG na Ukrajině (dříve také v Praze) kde je dále dekódován přijímači Tandberg nebo novějšími Ericsson. Následně je toto video převedeno pomocí Osprey karty případně přímo pomocí transport streamu (MPEG-TS UDP) do počítače jako vstupní signál pro Flash Media Live Encoder nebo FFMpeg.<sup>22</sup>

FMLE následně tato videa překóduje do požadovaného formátu. Pro video jde o kodek H.264, main profil, 1000Kbps, rozlišení 640x480, 30fps, klíčový snímek každých 5 sekund. Pro audio složku se používá kodek AAC, 128Kbps, stereo 48KHz.

Takto připravená videa jsou streamována pomocí Wowza serveru technologií HLS dále přes CDN Limelight až k zákazníkům v Severní americe, přesněji řečeno do jejich set-top-boxů.

U některých televizních kanálů se můžeme setkat s timeshiftem – tj. streamování s časovým posunem 5-7 hodin kvůli rozdílnosti časových pásem. Timeshift zajišťuje druhý server Wowza server pomocí funkce nDVR, který jednotlivé streamy uloží na disk a později je z takto uložených souborů pouští.

TV2Moro za pomocí timeshiftu streamuje 46 kanálů z celkových 98 a to pro více než 2000 uživatelů.

Pro projekt jsou použity z velké části standardní servery HP Proliant DL 360 G6 se šestijádrovým procesorem Xeon X5650 a 12 GB RAM pamětí. Na takovémto HW je možné současně enkódovat až 5 streamů s výše uvedenými parametry.

---

<sup>22</sup> dříve byl využíván Windows Media Encoder spolu s Window Media Serverem

## 8.3 Sky Go

Jedna z největších společností v Británii, která se zabývá také streamingem je společnost Sky. Podobně jako ČT provozuje satelitní televizní vysílání a zároveň nabízí některé pořady v podobě placeného streamingu pod názvem Sky Go.

Tato služba nabízí více než 40 live televizních kanálů a v podobě VoD přes 1000 různých videosoborů. Tyto streamy je možné shlédnout snad na všech platformách – od klasických počítačů, přes mobilní telefony s operačním systémem iOS nebo Android až po herní konzole Playstation a XBox.

Vzhledem k tomu, že se nepodařilo zajistit souhlas s uveřejněním větších podrobností o tomto projektu mohou být uvedeny pouze informace běžně dostupné.

Sky Go využívá týdně v průměru více než 1,4 miliónů unikátních uživatelů, přičemž najednou dokáže odbavit až 320 000 diváků. K tomu, aby dokázala odbavit tolik diváků najednou, potřebuje samozřejmě rozsáhlou síť CDN severů. Za tímto účelem je využívána síť Akamai zejména pro VoD obsah a Level3 pro live obsah.

## 9 Praktické testy

### 9.1 Cíl testování

Cílem testování byla realizace modelu streamingového serveru a následné porovnání účinnosti a výpočetní náročnosti jednotlivých technologií.

### 9.2 Vybrané řešení streamingového serveru

Pro účely testování byl zvolen streamovací server Wowza Media Server 3.6.3, build 8031 ve standardní instalaci na OS Debian GNU/Linux se zkušební („trial“) licencí. Tato licence má 30-denní platnost, více k licenční a cenové politice Wowza v kapitole 3.9, případně přímo na webových stránkách výrobce.

Toto řešení bylo vybráno pro jeho časté použití v praxi a také na základě podpory mnoha různých technologií.

### 9.3 Testovací prostředí

Jako streamovací server byl použit hardware desktopového typu s výpočetním výkonem 4 procesorového (Quad Core) AMD FX CPU a 8 GB RAM. Na tomto stroji byl nainstalován Wowza server běžící na 64-bitové platformě operačního systému Linux Debian verze 7.0 (Wheezy), více informací o použitém HW na přiloženém DVD.

Jako testovací klienti byly použity různé hardwarové typy počítačů, jako například Mac Mini, Macbook Pro, Dell Latitude E6410 , Dell Latitude E6230 a několik standardně sestavovaných počítačů.

Testování probíhalo na operačních systémech Linux Debian 7.0 (Wheezy) 64-bit, Mac OS X 10.9 Mavericks, Mac OS X 10.7 Lion, Windows 7 64bit, Windows 7 32bit.

Všechny počítače byly v průběhu testu připojeny pomocí standardních ethernetových (CAT 5E) kabelů do centrálního 1Gb switchu D-Link DGS-1100-16.

Jako náhledové monitorovací zařízení posloužila plasmová televize Panasonic TX-P42T30E propojená kabelem HDMI z Mac mini, případně z laptopu Dell.

### 9.4 Testovací oblasti

Testy byly vybrány s ohledem na dostupnou kvalitativní i kvantitativní hardwarovou náročnost. Server byl testován z těchto hledisek:

- výkonu – vytížení serveru v závislosti na počtu připojených klientů
- kvality – pozorování rozdílů mezi jednotlivými technologiemi

Pro účely testování bylo jako zdrojové video použito video s názvem Big Buck Bunny, volně dostupné na url: „www.bigbuckbunny.org“ ve formátu MOV H.264, AAC surround sound – Apple Quicktime Compactible v rozlišení 1920x1080 (16:9), 24fps s průměrným bitrate více než 9 Mbit/s (součástí přílohy - big\_buck\_bunny\_1080p\_h264.mov).

Toto zdrojové video bylo za použití Microsoft Expression Encoder překódováno do nižších datových toků, při použití kontejneru mp4 a kodeku H.264, přičemž audio bylo změněno z původního šestikanálového na pouze dvoukanálové (stereo) AAC.

Testovaná videa (mimo originálního) byla překódována do následujících bitových toků za použití variable bitrate:

- 5 Mbit/s (big\_buck\_bunny\_1080p\_h264\_5000.mp4)
  - 3 Mbit/s (big\_buck\_bunny\_1080p\_h264\_3000.mp4)
  - 1 Mbit/s (big\_buck\_bunny\_1080p\_h264\_1000.mp4)
- rozlišení zůstalo stejné a to 1920x1080.

Všechna použitá testovací videa jsou součástí příloženého DVD.

Tato videa byla použita jako zdroj pro streaming VoD, kdy dochází k větší zátěži serveru než při streamování live videa.

## 9.5 Testované technologie

Pro testování byly vybrány tři technologie. První dvě (RTMP a RTSP) jako zástupce klasického streamingu, hojně využívaných v set-top-boxech, a třetí, novější, technologie HDS, jako zástupce adaptivního streamingu.

### 9.5.1 RTMP

Jako první byla testována technologie RTMP, která má širokou podporu přehrávačů. Tato technologie byla bez problémů zpracována všemi testovanými desktopovými přehrávači: VLC (2.1.2), Mplayer (1.1.1) a QuickTime (7.7.4)

## 9.5.2 RTSP

RTSP technologie je také podporována všemi testovanými přehrávači. Implicitně používá RTSP pro streamování UDP, v některých přehrávačích je ale možno nastavit použití TCP.

## 9.5.3 HDS

HDS technologie není bohužel podporována ani jedním z testovacích standardních desktopových přehrávačů, proto byl v tomto případě použit dedikovaný flash přehrávač, který je součástí standardní instalace serveru Wowza.

## 9.6 Metodika testování

Pro docílení dostatečné zátěže serveru bylo nutné simulovat připojení více klientů, protože při streamování videa pouze pro jednoho klienta by byl rozdíl v jednotlivých technologiích zanedbatelný a tak by nedošlo k dosažení cíle testování.

K připojování jednotlivých klientů byl použit skript uvedený v příloze 1.

Tento skript měl během testování za úkol připojit se k testovacímu serveru určeným počtem klientů (proměnná „STREAM\_JOBS“) s určitým časovým zpožděním (proměnná STREAM\_DELAY).

Skript zinicizoval vícenásobné připojení klienta k serveru, a přijímání streamu, ale na koncových klientech nedocházelo k dekodování a zobrazování z důvodu snížení výpočetní náročnosti, aby bylo možno simulovat mnoho klientů.

K vizuální kontrole plynulosti a další kontrole kvality obrazu a zvuku posloužil náhledový klient.

Postupně byly provedeny testy simulace 40, 80 a 120, respektive včetně náhledového kontrolního streamu 41, 81 a 121 klientů.

K zajištění stavu kvality a plynulosti přehrávání posloužil právě náhledový monitor na kterém byl přehráván a sledován testovaný stream.

Při testování technologie HDS nemohla být popsána metodika aplikována, jelikož použité přehrávače nejsou kompatibilní s technologií HDS. Testování streamu tedy v tomto případě probíhalo za použití flash přehrávače, standardně dodávaného v rámci instalace Wowza serveru. Klienti byli pouštěni ručně na několika stanicích tak, aby všech 40 streamů běželo po dobu monitorování výsledků. Současně nebyl překročen počet předem stanovených

spuštěných přehrávačů na jednom počítači, aby bylo zabráněno jeho případnému přetížení. Dosažený počet klientů se velmi lišil v závislosti na hardware, neboť stream byl dekódován a zobrazován. Nejméně šlo o dva, nejvíce o 8 streamů na jednom zařízení. Z důvodu nedostupnosti většího počtu strojů byla technologie HDS testována pouze pro 40 klientů.

Pro každou technologii byly změřeny výsledky pro čtyři datové toky od 1 do více jak 9 Mbit/s s použitím kodeku H.264.

Stav zátěže serveru a provozu sítě byl monitorován pomocí několika monitorovacích nástrojů, popsanych v dalších kapitolách:

- JConsole
- WMSpanel
- Darkstat
- TOP

Naměřené výsledky pro jednotlivé protokoly jsou následně zpracovány viz souhrn.

## 9.6.1 JConsole

JConsole je součástí instalace balíku Sun JDK, pouze je potřeba ji v konfiguračním souboru Wowzy ([instalační adresář Wowza]/conf/Server.xml) povolit změnou parametru Enable na „true” a následným restartem Wowza Media Serveru.

```
<Root>
  <Server>
    <JMXRemoteConfiguration>
      <Enable>true</Enable>
      <IpAddress>localhost</IpAddress>
      <RMIServerHostName>localhost</RMIServerHostName>
```

Následným příkazem:

```
service:jmx:rmi://localhost:8084/jndi/rmi://localhost:8085/jmxrmi
```

JConsoli spustíme a připojíme se s defaultním jménem a heslem (admin / admin)<sup>23</sup> k aktuálním statistikám Wowza serveru.

Na následujícím obrázku je vidět prakticky příklad výstupu Jconsole, kde můžeme vidět graf „Heap Memory Usage“ tj. využití pracovní paměti, graf s počtem vláken „Threads“, graf s počtem načtených tříd „Classes“ a graf zobrazující vytížení procesoru „CPU“.

---

<sup>23</sup> Samozřejmě pouze pokud nebylo výchozí heslo změněno

## 9.6.2 WMSPanel

WMSPanel je rozšíření pro Wowzu, které umožňuje monitorování provozu serveru. Tato služba je po registraci zdarma na vyzkoušení po dobu 2 týdnů, poté je nutno zaplatit za jednotlivé licence pro každou instanci Wowza serveru. WMSPanel se nainstaluje na server, kde je instalace Wowzy a od té doby začne odesílat data na agregační server, kde je možné po přihlášení sledovat statistiky. Na výběr máme grafy jak celkového provozu serveru, tak počet aktuálně streamovaných videí nebo také statistiky pro streamování jednotlivých kvalit či souborů s videem.

## 9.6.3 Darkstat

Darkstat je volně dostupný monitorovací nástroj pro provoz síťových rozhraní. Neumí sice monitorovat přímo Wowzu jako takovou, ale pro porovnání celkového síťového provozu je určitě vhodným nástrojem.

Tento software byl nainstalován v podobě balíčku pro Debian ve standardní verzi dostupné pro stabilní větev v době testování – Debian stable/Wheezy.

Výstup z Darkstatu je možno prohlížet pomocí webového rozhraní, kde jsou zobrazeny grafy provozu síťového rozhraní za posledních 60 sekund, 60 minut, 24 hodin a 31 dní.

## 9.6.4 Top

Top je částečně interaktivní program zobrazující real-time statistiky, týkající se provozu na UNIX-like serveru, v našem případě na serveru typu Linux.

Top používá pro získávání dat standardní rozhraní OS Linux. Zobrazovaná data jsou sbírána přímo jádrem operačního systému Linux a jsou tak vysoce relevantní, neboť v rámci serveru neexistují data s vyšší přesností.

Mezi zobrazované informace patří kromě výpisu běžících procesů a systémových prostředků jimi alokovaných taktéž globální a průměrné hodnoty využití a zátěže systému jako celku.



## 9.7 Testování

Testování probíhalo postupně pro jednotlivé protokoly, přičemž RTMP protokol byl testován jako první, a byl tudíž zvolen jako referenční pro stanovení počtu připojovaných klientů.

### 9.7.1 RTMP

Pro určení počtu klientů bylo testováno video s nejvyšším datovým tokem, tj. video s více jak 9Mbit/s. Pro zjednodušení bude toto video dále označeno jako video s průměrným datovým tokem 9Mbit/s.

Nejdříve bylo testováno 40 klientů, které server stihl odbavit bez jakýchkoliv problémů, což bylo vidět jak na zátěži, tak i na náhledovém monitorovacím streamu. Přehrávané video bylo bez známek artefaktů a perfektně plynulé.

Při testování 80 klientů byl již procesor serveru přetížen a na monitorovacím klientovi docházelo k výpadkům plynulosti jak ve video, tak i v audio složce.

Pokus o připojení 120 klientů měl za následek pád streamovacího serveru Wowza.

Při testování sekvencí s nižším datovým tokem, tj. 5, 3 a 1 Mbit/s již nebyl problém ani při připojení 120 klientů.

### 9.7.2 RTSP

Při přehrávání protokolu RTSP mají použité přehrávače (VLC, Mplayer, Quicktime) problémy při streamování vyššího datového toku (5 a 9 Mbit/s) a dochází k častým chybám v obraze. Chyby však nejsou ovlivněny počtem jednotlivých klientů na jednom počítači, což bylo ověřeno tím, že z jednoho HW (byly vyzkoušeny všechny jednotlivé počítače s různými operačními systémy) byl puštěn pouze jeden přehrávací klient. Tím bylo vyloučeno přetížení streamovacího serveru, případně i přetížení sítě.

Pro zmenšení chyb ve streamu byl proveden pokus o překódování na jiný kodek i se změnou parametrů (zvětšení počtu klíčových snímků, změna odhadu kvanzice pohybu). Počet chyb se však zmenšil jen minimálně, drtivá většina problémů během přehrávání přetrvala.

Ke značnému zlepšení došlo při použití protokolu TCP namísto standardního. Toho bylo dosaženo pomocí přepínače „-rtsp-stream-over-tcp“ v přehrávači Mplayer. Výsledná kvalita se velmi blížila ideální kvalitě streamu s výjimkou řídké ztráty plynulosti přehrávání.

Problém se dále objevil v případě napojení dvou klientů z jednoho HW na stejný stream, kde valně nepomohlo ani přepnutí na protokol TCP.

Jelikož RTSP vytváří stream pro audio i video odděleně a Mplayer neumožňuje přepínač „-dumpstream“ a zároveň muxování streamu do jednoho souboru, musel být pro příjem streamu kvůli simulaci dostatečného počtu klientů použit ffmpeg.

Jelikož 120 klientů při datovém toku 9 Mbit/s nebylo pro protokol RTMP průchozích (došlo k pádu serveru) a nebyla tak pro tuto konfiguraci získána relevantní data, bylo RTSP testováno pro datový tok 9 Mbit/s v maximálním počtu 80 klientů.

V průběhu tohoto testování bylo možné pozorovat na monitorovacím klientu větší problémy s plynulostí přehrávání, přičemž třem klientům se ani nepodařilo k serveru připojit.

Z důvodu vyloučení vlivu rychlosti úložiště serveru, kde byly jednotlivé video soubory uloženy, byla provedena revalidace pro 9 Mbit/s video s 80 klienty tak, že streamovaný video soubor byl celý přesunut do paměti RAM, odkud byl následně streamován. Situace se však opakovala a výsledek byl velmi podobný – nepřipojili se všichni klienti a problémy s přehráváním na monitorovacím klientovi přetrvávaly.

Pro ukázkou s datovým tokem 5 Mbit/s bylo již testováno i 120 klientů, ale na monitorovacím klientovi byly pozorovány problémy s plynulostí při startu a ojediněle i v průběhu přehrávání. Při testování této kvality nebyl pozorován již žádný problém při 40 a ani při 80 klientech.

Při testech s nižším datovým tokem 3 a 1 Mbit/s nebyl problém ani při počtu 120 klientů.

### **9.7.3 HDS**

HDS technologie, jak již bylo zmíněno dříve, musela být testována částečně pozměněnou metodikou. Z důvodu nefunkčnosti testovacího Flash playeru pod systémem Linux bylo odebrání jednotlivých streamů startováno na operačních systémech MAC OS X a Windows manuálně. Z důvodu nedostupnosti většího počtu HW a také časové náročnosti manuálního spouštění přehrávačů byl tento test proveden pouze pro počet 40 klientů.

Pro stream s datovým tokem 1 Mbit/s bylo přehrávání bez jakýchkoliv problémů, pro video s 3Mbit/s v počtu blízcímu se 40 klientů již docházelo k mírným problémům s plynulostí přehrávání. To však nebylo způsobeno na straně serveru (jak je možné vidět na následujícím obrázku kde je zobrazen výstup z JConsole), nýbrž nedostatkem výkonu pro dekódování streamu na klientských stanicích. Při testech s videem o datovém toku 5 Mbit/s již

docházelo k významnějším problémům s plynulostí a to už od počtu 30 konkurentních klientů. Toto bylo opět způsobeno nedostatečným počtem a výkonem klientských stanic, ne vinou serveru. Z tohoto důvodu nebyl prováděn test pro kvalitu 9Mbit/s.

## 9.8 Srovnání

Srovnávání bylo rozděleno na jednotlivé kategorie z hlediska:

- výpočetní náročnosti na straně serveru v závislosti na počtu klientů
- výpočetní náročnosti na straně serveru v závislosti na bitovém toku a počtu klientů
- bitového toku na technologii při 40 klientech.

K porovnání byly zvoleny výstupní hodnoty vytížení procesoru a paměti pomocí příkazu Top.

Pro vytížení procesoru je uvedena první (aktuální) hodnota „load average“ vypovídající o průměrném vytížení procesoru všemi procesy za poslední minutu. Vzhledem k tomu, že v průběhu testů na serveru kromě základních věcí potřebných ke streamování běželi pouze monitorovací aplikace (vždy stejné), je tato hodnota brána jako vypovídající hodnotou zatížení výpočetního výkonu.

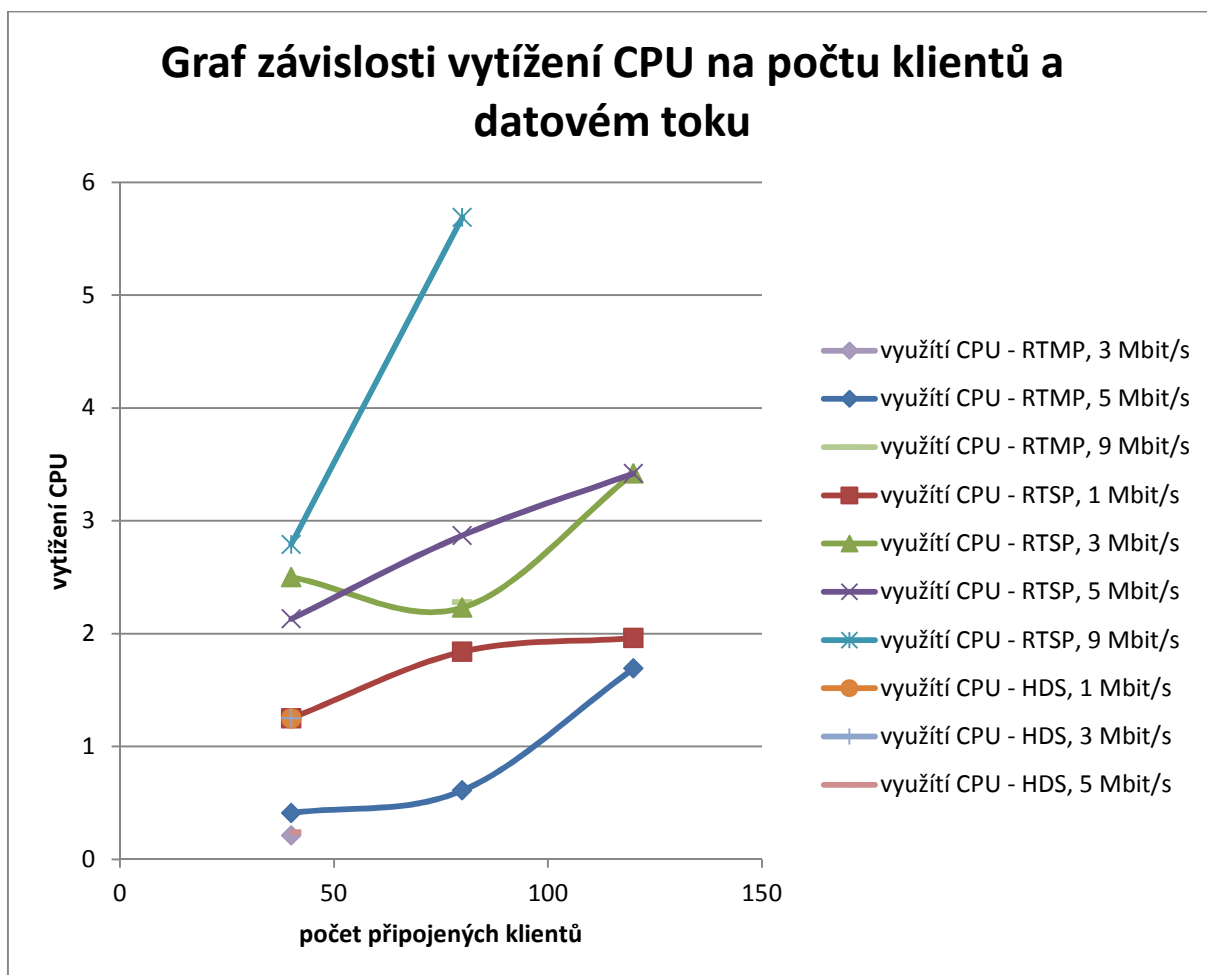
Hraniční hodnotou pro čtyřjádrový procesor, kdy je testovaný počítač plně vytížen je hodnota 4, která nebyla mimo ojedinělé případy (RTSP 9 Mbit/s) naměřena. I když v případě testu technologie RTMP se 120 klienty a videem s datovým tokem přes 9 Mbit/s tato hodnota mohla být taktéž dosažena, došlo dříve k pádu streamovací aplikace.

Dále byla pro porovnání využití paměti zvolena hodnota „RES“ (resident size), která ukazuje skutečnou velikost paměti využívané procesy. Tato hodnota je také součástí výstupu příkazu TOP.

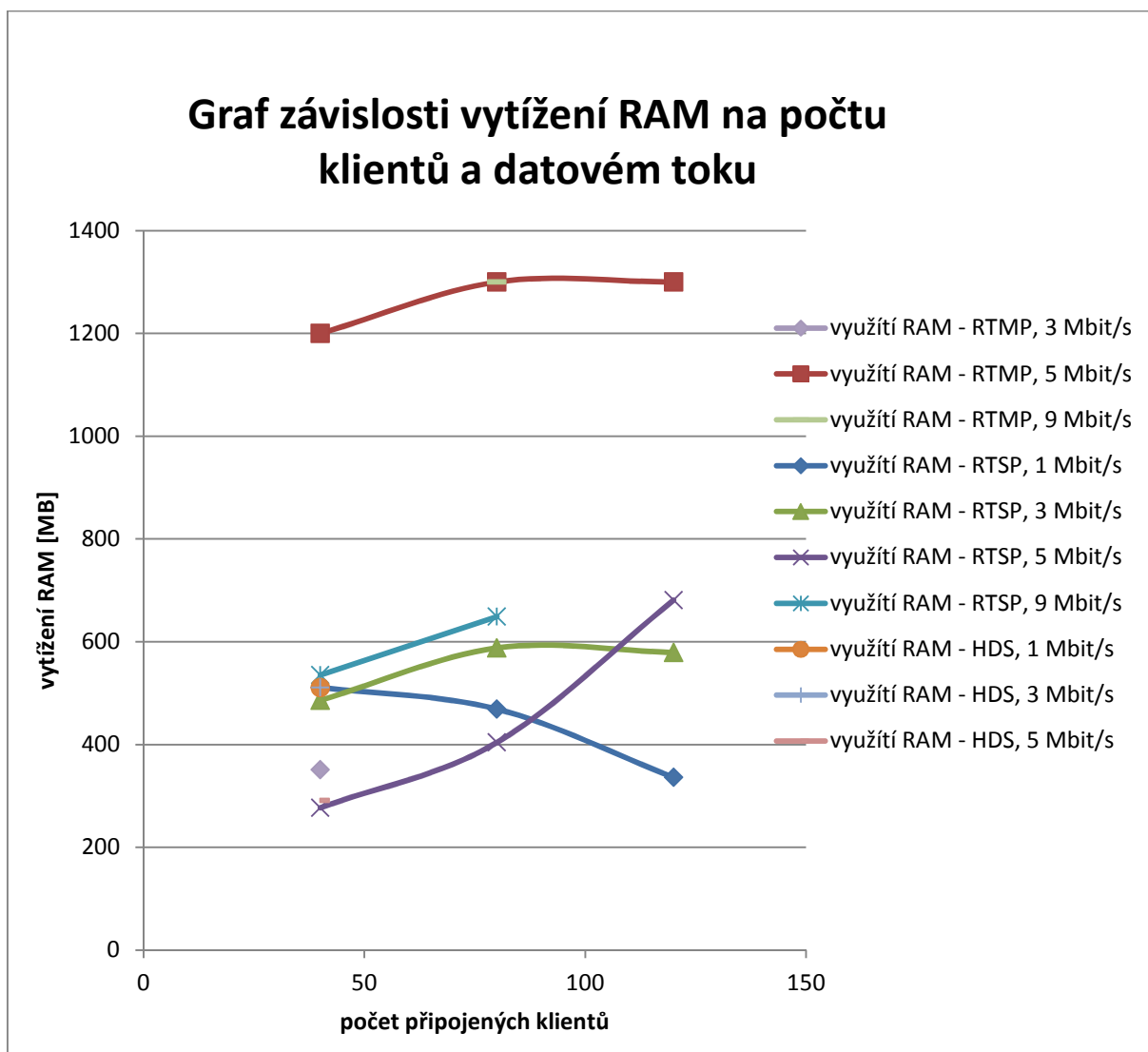
### 9.8.1 Grafické zpracování

Výpočetní zátěž serveru byla na počtu klientů dle předpokladů víceméně přímo závislá. V tabulce uvedené v příloze je možné porovnat závislosti počtu připojených klientů na výkonu pro technologie RTMP a RTSP, pro technologii HDS z výše uvedených důvodů nebylo možné tato data změřit.

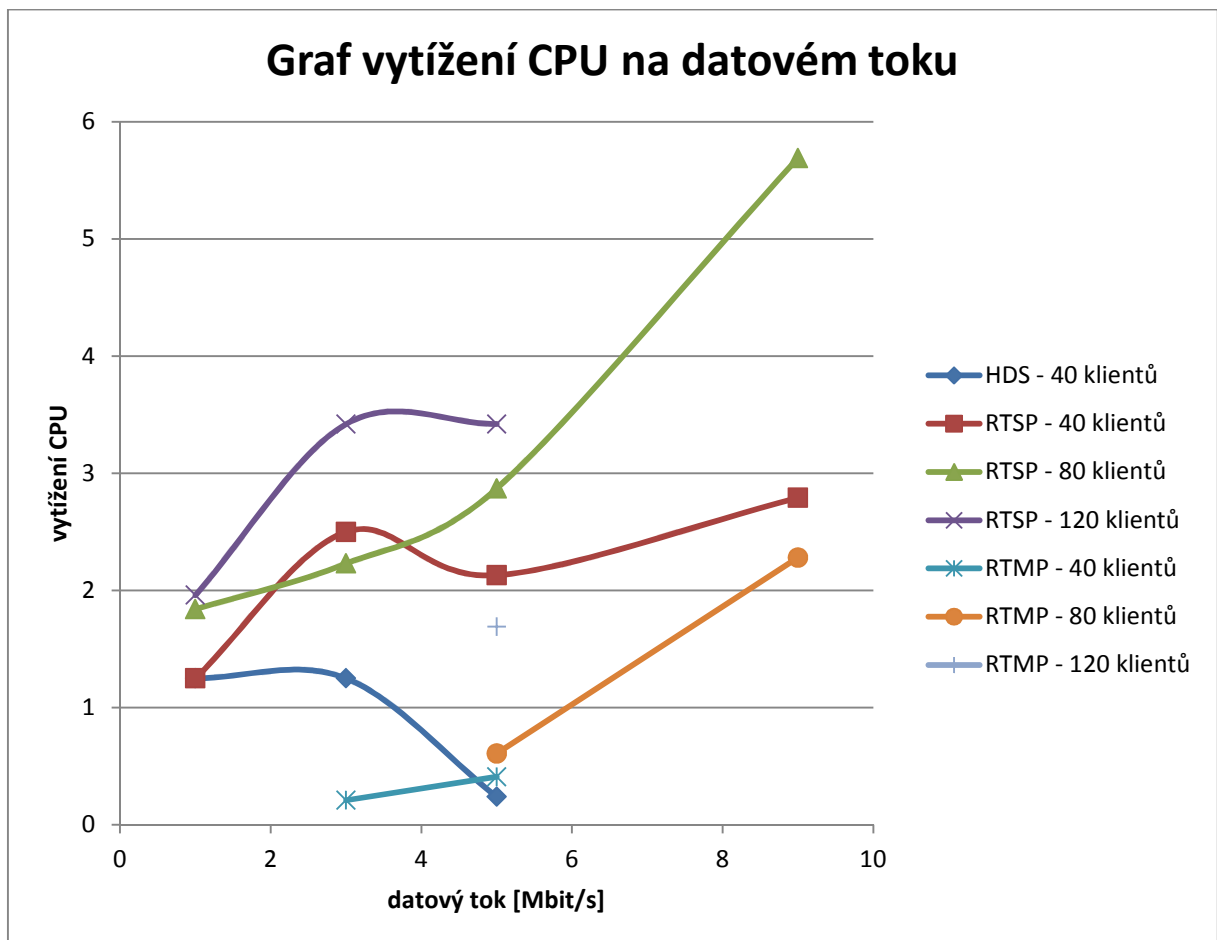
Zde jsou graficky znázorněny naměřené závislosti:



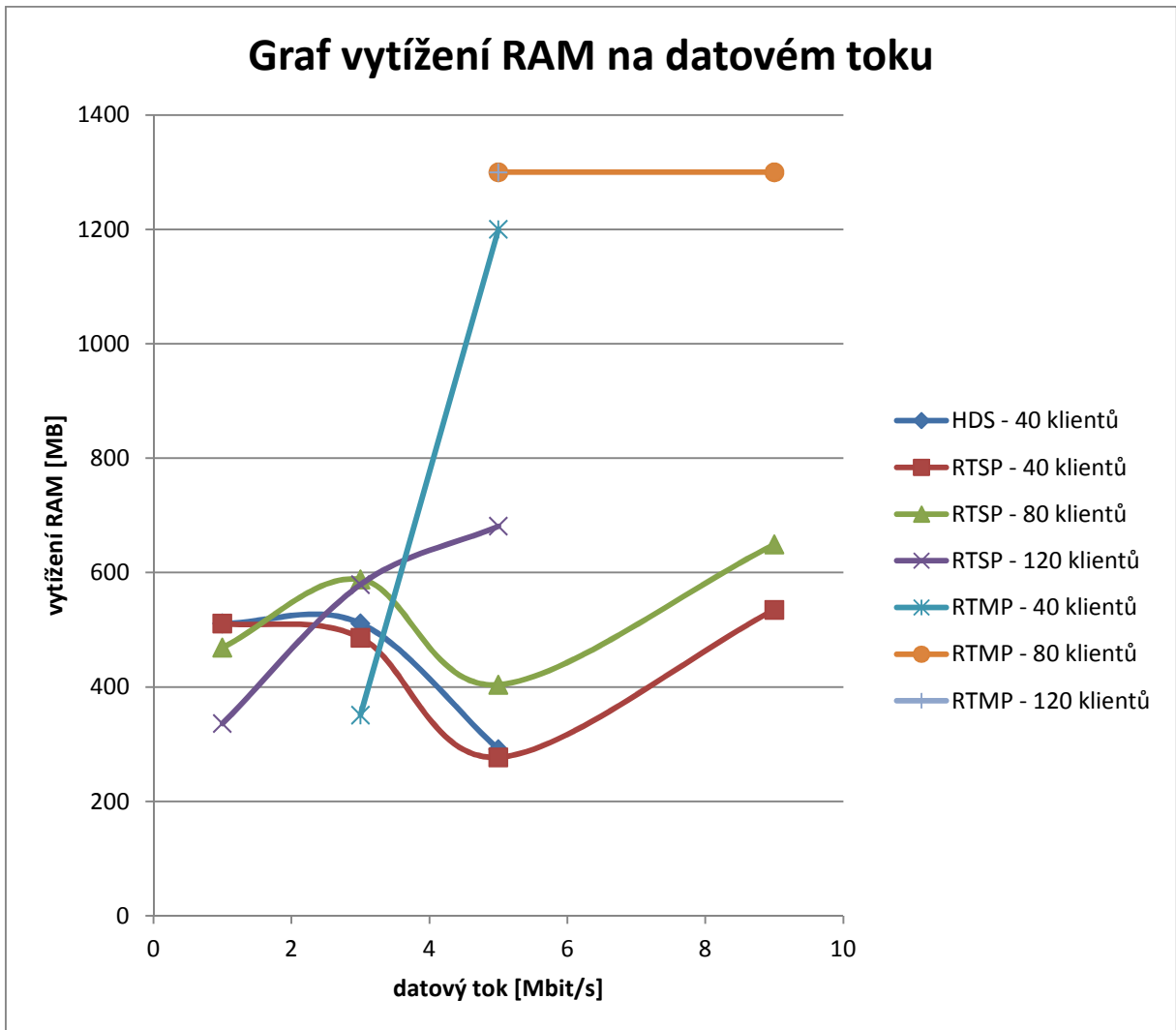
Graf 9.1: Graf závislosti vytížení CPU na počtu klientů a datovém toku



Graf 9.2: Graf závislosti vytížení RAM na počtu klientů a datovém toku



Graf 9.3: Graf vytížení CPU na datovém toku



Graf 9.4: Graf vytížení RAM na datovém toku

## 9.9 Vyhodnocení výsledků

Výsledky výpočetní náročnosti serveru na počtu klientů byly dle předpokladů pro každou technologii jiné.

Při počtu 40 klientů vyšla nejlépe (t.j. nejméně zatěžující) technologie HDS při datovém toku 5Mbit/s. U technologie RTMP za stejných podmínek bylo vytížení procesoru jen o něco málo větší, zato však využití paměti bylo téměř dvojnásobné. Naopak tomu bylo u technologie RTSP, kde byla hodnota využití CPU zvýšena téměř na desetinásobek, hodnota paměti však byla o něco málo menší než u HDS.

S přibývajícím počtem klientů většinou systémové nároky severu rostly, za zmínku však stojí technologie RTSP, která při datovém toku 1Mbit/s potřebovala menší paměť při větším počtu uživatelů, ale o to strměji narostla křivka pro využití procesoru.

V závislosti na bitovém toku nejlépe vyšla technologie HDS, která s narůstající kvalitou videa vykazovala menší nároky na výpočetní výkon a při datovém toku 5 Mbit/s nám vyšla v porovnání nejlépe.

Z výše uvedených grafů je zřejmé, že výpočetní náročnost streamingového řešení je závislá na počtu uživatelů, ale také na datovém toku streamu.

Protokoly RTSP a RTMP jsou vhodné pro streamování menších datových toků a jejich využití je spíše v set-top-boxech, pro které se nejčastěji používají videa s datovým tokem okolo 1,5Mbit/s.

Pro vyšší kvality je vhodnější technologie HDS, klade menší nároky na server.

Všechny naměřené výstupy jsou součástí příloženého DVD. Zde nejsou uvedeny proto, že nevykazují přesné hodnoty, které by se daly porovnávat z důvodu jejich nestálého průběhu či pomalejší odezvy na aktuální vytížení (v případě WMS konzole).

Testy byly provedeny tak, aby se co nejvíce podobaly realistické zátěži a naměřené výsledky tak co nejvíce odpovídaly reálnému provozu.



## 10 Závěr

V této práci jsou stručně popsány procesy, které se dějí nejen při samotném streamingu videa, ale také postupy, které jsou aplikovány nejen na samotný streaming.

Základem streamování je zmenšení datového toku videa, což mají za úkol různé komprimační algoritmy v práci popsané. Komprese videa je nejdůležitější část, při které se do výsledného audiovizuálního obsahu dostává nejvíce zkreslujících artefaktů.

Kompresa videa je důležitá i v dnešní době, kdy se rychlost a propustnost síťových technologií a internetu mnohonásobně zvýšila. Vývoj audiovizuální techniky se samozřejmě rovněž nezastavil – to, co se ještě před pár lety zdálo být kvalitním zdrojem pro pořizování videa, v dnešní době už neplatí. To má ale za následek zvyšující se nároky na kapacitu pořizovaných záznamů, které jsou většinou mimo možnosti průměrných zařízení a kapacity internetových připojení.

V práci byly následně podrobně rozebrány celý streamovací řetězec a jednotlivé aplikace pro streamování, na jehož konec můžeme zařadit i metody pro hodnocení kvality obsahu.

V praktické části byly otestovány tři technologie a jejich výpočetní náročnost. Z výsledků je zřejmé, že nejde zvolit jen a pouze jednu technologii jako správnou nebo nejlepší, protože závisí na druhu použití. Jinou technologii můžeme zvolit v případě, kdy klient disponuje dostatečně dobrým hardware a jinou zase v případě kdy zvolíme přehrávač v podobě set-top-boxu, který nedisponuje takovým výkonem jako osobní počítač.

## 11 Použité zkratky

AAC	Advanced Audio Coding
AVC	Advanced Video Coding
CDN	Content Delivery Network
CPU	Central Processing Unit
DASH	Dynamic Adaptive Streaming over HTTP
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DRM	Digital Rights Management
DSCQS	Double Stimulus Continuous Quality Scale
DSIS	Double Stimulus Impairment Scale
DST	Discrete Sine Transform
DVB	Digital Video Broadcasting
FMLE	Flash Media Live Encoder
GOP	Group Of Pictures
HDS	HTTP Dynamic Streaming
HEVC	High Efficiency Video Coding
HLS	HTTP Live Streaming
HVS	Human Visual System
MPD	Media Presentation Description MSE
RLC	Run-Length Coding
RTMP	Real Time Messaging Protocol
RTSP	Real Time Streaming Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UHD	Ultra High Definition
VLC	Variable Length Coding
VoD	Video on Demand

## 12 Seznam použité literatury a elektronických zdrojů

- [1] [1] Wu, H.R., Rao, K.R. *Digital Video Image Quality and Perceptual Coding, Signal Processing and Communications*. Boca Raton: CRC Press, 2006. ISBN 978-0-8247-2777-2.
- [2] Winkler, S. *Digital Video Quality: Vision Models and Metrics*. Chichester: Wiley, 2005. ISBN 0-470-02404-6.
- [3] Tomaidēs, P. Streaming – konec televize? [online]. In: *PORT*. Česká televize, 2008. [vid. 25.07.2012]. Dostupné z: <http://www.ceskatelevize.cz/porady/10121359557-port/technosfera/252-streaming-konec-televize/>
- [4] Co je to streaming? [online]. [vid. 25.07.2012]. Dostupné z: <http://streamdownload.okamzite.eu/co-je-to-streaming/>
- [5] Smith, Steven W. *The Scientist and Engineer's Guide to Digital Signal Processing* [online]. San Diego: California Technical Publishing, 1997. ISBN 0-9660176-3-3. [vid. 25.07.2012]. Dostupné z: [www.dspguide.com/](http://www.dspguide.com/)
- [6] Antonini, M., Barlaud, M., Mathieu, P., Daubechies, I. Image coding using wavelet transform. *IEEE Transactions on Image Processing*. 1992, Vol. 1, Issue 2, 205-220. ISSN 1057-7149.
- [7] Khayam, S.A. *The Discrete Cosine Transform (DCT): Theory and Application* [online]. Michigan: Michigan State university, 2003. [vid. 25.7.2012]. Dostupné z: <http://ebookbrowsee.net/the-discrete-cosine-transform-dct-theory-and-application-by-syed-ali-khayam-march-10th-2003-pdf-d326144520>
- [8] Tinka, L.: *Jak funguje JPEG – princip pod lupou*. [online]. [vid. 25.7.2012]. Dostupné z: [http://libca.cz/pub/libor/web-funny/documentation/files/Jak\\_funguje\\_JPEG.pdf](http://libca.cz/pub/libor/web-funny/documentation/files/Jak_funguje_JPEG.pdf)

- [9] Říčný, V. *Videotechnika – Přednášky*. 4. upravené vydání. Brno: VUT v Brně, 2006. ISBN 80-214-3225-X
- [10] ISO/IEC 13818-2:2000. *Information technology - Generic coding of moving pictures and associated audio information: Video*. ICS: 35.040. Ed. 2, 2000.
- [11] ISO/IEC 14496-2:1999. *Information Technology-Coding of audio-visual objects-Part 2:Visual*. Ed. 1, 1999.
- [12] ISO/IEC 14496-10:2005. *Information Technology-Coding of audio-visual objects-Part 10:Advanced Video Coding*. Ed. 1, 2005.
- [13] International Telecommunication Union [online]. *Video coding for low bit rate communication ITU-T Recommendation H.263*. 1998. [vid. 25.07.2012]. Dostupné z: <http://www.itu.int/rec/T-REC-H.263-199802-S/en>
- [14] International Telecommunication Union [online]. *Advanced video coding for generic audiovisual services ITU-T Recommendation H. 264*. 2005. [vid. 25.07.2012]. Dostupné z: <http://www.itu.int/rec/T-REC-H.264-200503-S/en>
- [15] *Theora I specification. Xiph.org Foundation* [online]. Theora.org. [vid. 25.07.2012]. Dostupné z: <http://www.theora.org/doc/Theora.pdf>
- [16] *Microsoft Windows Media webpage* [online]. Microsoft Corporation. [vid. 25.07.2012]. Dostupné z: <http://windows.microsoft.com/en-US/windows/products/windows-media>
- [17] *Real Networks webpage* [online]. RealNetworks, Inc. [vid. 25.07.2012]. Dostupné z: <http://www.realnetworks.com/>
- [18] Li, S., Yu, Jiang, G., Choi, T., Kim, Y. Approaches to H.264-Based Stereoscopic Video Coding, *Proceedings of the Third International Conference on Image and Graphics (ICIG '04)*. 2004, 365-368. ISBN 0-7695-2244-0.

- [19] Ozer, J. What is Streaming. In: *Streaming Media*. StreamingMedia.com, 2011. [vid. 25.07.2012]. Dostupné z: <http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-Streaming-74052.aspx>
- [20] *Adobe Systems webpage* [online]. Adobe Systems, Inc. [vid. 25.7.2012]. Dostupné z: <http://www.adobe.com>
- [21] *VideoLan Organization webpage* [online]. VideoLan. [vid. 25.7.2012]. Dostupné z: <http://www.videolan.org>
- [22] *Apple webpage* [online]. Apple Computer, Inc. [vid. 25.7.2012] Dostupné z: <http://www.apple.com>
- [23] Wang, Z., Sheikh, H.R., Bovik, A.C. Objective video quality assessment. In: B. Furht and O. Marqure. *The Handbook of Video Databases: Design and Application* [online]. CRC Press, 2003. 1041-1078. ISBN 978-0-203-48986-4. [vid. 25.7.2012]. Dostupné z: [https://ece.uwaterloo.ca/~z70wang/publications/QA\\_hvd\\_bookchapter.pdf](https://ece.uwaterloo.ca/~z70wang/publications/QA_hvd_bookchapter.pdf)
- [24] Lederer, S., Mueller, Ch., Rainer, B., Timmerer, Ch., Hellwagner, H. Adaptive Streaming over Content Centric Networks in Mobile Networks using Multiple Links. In: *Communications Workshops (ICC), 2013 IEEE International Conference on* [online]. Budapest: IEEE, 2013. [vid. 25.7.2012]. Dostupné z: [http://www-itec.uni-klu.ac.at/bib/files/ICC2013\\_Mobile\\_DASHoverCCN.pdf](http://www-itec.uni-klu.ac.at/bib/files/ICC2013_Mobile_DASHoverCCN.pdf)
- [25] Vcodex Limited [online]. *HEVC: An Introduction to High Efficiency Video Coding [SUMMARY VERSION]*. 2013. [vid. 25.7.2012]. Dostupné z: <http://www.vcodex.com/images/uploaded/342512928230717.pdf>
- [26] *RTP Payload Format for VP8 Video draft-ietf-payload-vp8-10* [online]. IETF. [vid. 25.7.2012]. Dostupné z: <http://tools.ietf.org/html/draft-ietf-payload-vp8-10>

- [27] Bovik, A. *The Essential Guide to Video Processing*. 2<sup>nd</sup> edition. San Diego:Elsevier, 2009. 778 stran. ISBN 978-0-12-374456-2.
- [28] ISO/IEC 23008-2. *High Efficing Video Coding ITU-T H.265*. Ed. 1. International Telecommunication Union, April, 2013. Dostupné z:  
<http://handle.itu.int/11.1002/1000/11885-en?locatt=format:pdf>
- [29] *Helix* [online]. RealNetworks Inc. [vid. 25.7.2012]. Dostupné z:  
<http://www.realnetworks.com/helix/index.aspx>
- [30] RealNetworks Inc. [online]. *Helix Broadcaster: High-performance media encoding and delivery*. 2013. [vid. 25.7.2012]. Dostupné z:  
<http://www.realnetworks.com/uploadedFiles/helix/resource-library/Datasheet-Helix-Broadcaster.pdf>
- [31] RealNetworks Inc. [online]. *Helix Media Deliver Platform Integration Guide*. 2012. [vid. 25.7.2012]. Dostupné z:  
[http://helixproducts.real.com/docs/server15/HelixServer\\_IntegrationGuide\\_v150\\_GA.pdf](http://helixproducts.real.com/docs/server15/HelixServer_IntegrationGuide_v150_GA.pdf)
- [32] Ferreira A. Optimizing Windows Media Services. In: *Windows Media*. Microsoft Corporation, 2005. [vid. 25.7.2012]. Dostupné z:  
[http://www.microsoft.com/windows/windowsmedia/howto/articles/optimize\\_web.aspx](http://www.microsoft.com/windows/windowsmedia/howto/articles/optimize_web.aspx)
- [33] *Windows Media Services* [online]. Microsoft Corporation. [vid. 25.7.2012]. Dostupné z: <http://technet.microsoft.com/library/ee822833%28WS.10%29.aspx>
- [34] *Windows Media Encoder* [online]. Microsoft Corporation. [vid. 25.7.2012]. Dostupné z: <http://technet.microsoft.com/en-us/library/bb676137.aspx>
- [35] Zambelli, A. IIS Smooth Straming Technical Overview. In: *Microsoft Download Centre*. Microsoft Corporation, 2009. [vid. 25.7.2012]. Dostupné z:  
<http://www.microsoft.com/en-us/download/details.aspx?id=17678>

- [36] *Announcing Expression Encoder 2* [online]. Microsoft Corporation. [vid. 25.7.2012].  
Dostupné z:  
<http://blogs.msdn.com/b/expressionencoder/archive/2008/03/06/8064390.aspx>
- [37] *What's new in Expression Encoder 3* [online]. Microsoft Corporation. [vid. 25.7.2012]. Dostupné z:  
<http://blogs.msdn.com/b/expressionencoder/archive/2009/07/10/9828866.aspx>
- [38] *Supported file formats* [online]. Microsoft Corporation. [vid. 25.7.2012]. Dostupné z:  
<http://msdn.microsoft.com/en-us/library/cc294687%28v=expression.40%29.aspx>
- [39] Apple Computer, Inc. [online]. *Quick Time Streaming Server, Darwin Streaming Server*. 2002. [vid. 25.7.2012]. Dostupné z:  
[http://manuals.info.apple.com/MANUALS/0/MA579/en\\_US/QuickTimeStreamingServerAdminGuide.pdf](http://manuals.info.apple.com/MANUALS/0/MA579/en_US/QuickTimeStreamingServerAdminGuide.pdf)
- [40] *Using HTTP Live Streaming* [online]. Apple Computer, Inc. [vid. 25.7.2012].  
Dostupné z:  
[https://developer.apple.com/library/ios/documentation/networkinginternet/conceptual/streamingmediaguide/UsingHTTPLiveStreaming/UsingHTTPLiveStreaming.html#//apple\\_ref/doc/uid/TP40008332-CH102-SW1](https://developer.apple.com/library/ios/documentation/networkinginternet/conceptual/streamingmediaguide/UsingHTTPLiveStreaming/UsingHTTPLiveStreaming.html#//apple_ref/doc/uid/TP40008332-CH102-SW1)
- [41] Adobe Systems Incorporated [online]. *Introducing Adobe Media Server 5 white paper*. 2012. [vid. 20.7.2013]. Dostupné z:  
<http://www.adobe.com/content/dam/Adobe/en/products/ams/pdfs/ams5-intro-wp.pdf>
- [42] Wowza Media Systems, LLC [online]. *Wowza Media Server – Overview*. 2013. [vid. 20.7.2013]. Dostupné z:  
<http://www.wowza.com/uploads/images/WowzaMediaServer3-Product-Overview.pdf>
- [43] Wowza Media Systems, LLC [online]. *Wowza Media Server 3 – User's Guide*. 2013. [vid. 20.7.2013]. Dostupné z:  
[http://www.wowza.com/resources/WowzaMediaServer\\_UsersGuide.pdf](http://www.wowza.com/resources/WowzaMediaServer_UsersGuide.pdf)

- [44] *FFmpeg* [online]. FFmpeg. [vid. 25.8.2013]. Dostupné z: <http://www.ffmpeg.org>
- [45] Korbelt, F. *FFmpeg Basics: Multimedia handling with a fast audio and video encoder*. 2012. 215 stran. ISBN 1479327832.
- [46] Thang, T.C., Ho, Q.-D., Kang, J.W., Pham, A.T. Adaptive Streaming of Audiovisual Content Using MPEG DASH. *Consumer Electronics*. 2012, Vol. 58, Issue: 1, 78-85. ISSN 0098-3063.
- [47] Wang, Zhou, Sheikh, Hamid R., Bovik, Alan C. Objective video quality assessment. *The Handbook of Video Databases: Design and Applications* [online]. 2003 [vid. 25.9.2013]. Dostupné z: [https://ece.uwaterloo.ca/~z70wang/publications/QA\\_hvd\\_bookchapter.pdf](https://ece.uwaterloo.ca/~z70wang/publications/QA_hvd_bookchapter.pdf)
- [48] *Dynamic Adaptive Streaming over http (DASH)* [online]. SlideShare, Inc. [vid. 25.9.2013]. Dostupné z: <http://www.slideshare.net/christian.timmerer/dynamic-adaptive-streaming-over-http-dash>
- [49] Stockhammer, T. Dynamic Adaptive Streaming over HTTP – Design Principles and Standards. In: *MMSys '11 Proceedings of the second annual ACM conference on Multimedia systems* [online]. Santa Clara: MMSYS, 2011. [vid. 25.9.2013]. Dostupné z: [http://www.w3.org/2010/11/web-and-tv/papers/webtv2\\_submission\\_64.pdf](http://www.w3.org/2010/11/web-and-tv/papers/webtv2_submission_64.pdf)



## 13 Přílohy

### 13.1 Příloha 1 – použitý script pro testování

```
STR_SERVER=${STREAM_SERVER:-"172.17.2.1:1935"}
STR_DUMP=${STREAM_DUMP:-""} # neprázdný = ulož
STR_SAVEFILE=${STREAM_SAVEFILE:-"/dev/null"} # jméno souboru, default: zahod'
PLAYER_RTMP="mplayer"
PLAYER_RTSP_VISUAL="mplayer -rtsp-stream-over-tcp -framedrop" # UDP->TCP, kvalita+, framedrop
kvůli přehrávání
PLAYER_RTSP_DUMP="ffmpeg -i"
STR_PROTO=${STREAM_PROTOCOL:-"rtmp"}
STR_PATH=${STREAM_PATH:-"vod"} # cesta k Wowza 'stream aplikaci'
STR_MEDIA=${STREAM_MEDIA:-"big_buck_bunny_1080p_h264_5000.mp4"}
STR_JOBS=${STREAM_JOBS:-"1"} # počet spouštěných procesů
STR_DELAY=${STREAM_DELAY:-"3"} # pauza mezi spouštěnými procesy

PLAYER=""
EXTRA=""

if [ "$STR_PROTO" = "rtmp" ]; then
    PLAYER="$PLAYER_RTMP"
    STR_MEDIA="mp4:$STR_MEDIA"
    if [ -n "$STR_DUMP" ]; then
        EXTRA="-dumpstream -dumpfile $STR_SAVEFILE"
    fi
elif [ "$STR_PROTO" = "rtsp" ]; then
    PLAYER="$PLAYER_RTSP_VISUAL"
    if [ -n "$STR_DUMP" ]; then
        PLAYER="$PLAYER_RTSP_DUMP" # RTSP je multistream (vid+aud), nejde uložit přes '-
dumpstream'
        EXTRA="-acodec copy -vcodec copy -f mp4 -y $STR_SAVEFILE"
        STR_SERVER="@$STR_SERVER"
    fi
else
    echo "Neznámý protokol"
    exit 1
fi
```

```

ADDR="${STR_PROTO}://${STR_SERVER}/${STR_PATH}/${STR_MEDIA}"

echo "Adresa je $ADDR, pouštět budu $STR_JOBS procesů s pauzou $STR_DELAY"
JOB=0;
FNEXT=""; # přípona k souboru, pokud se jich má ukládat více
while [ $STR_JOBS -gt $JOB ]; do
    JOB=$((JOB + 1))
    if [ -n "$STR_DUMP" ] && [ $STR_JOBS -gt 1 ] && [ "$STR_SAVEFILE" != "/dev/null" ]; then
        # když se má ukládat a zároveň se má pustit více procesů a nemají zahazovat, nastav příponu
        FNEXT=".$JOB"
    fi
    echo "Spouštím proces číslo $JOB..."
    echo $PLAYER $ADDR $EXTRA      ### DEBUG ###
    xterm -e $PLAYER $ADDR $EXTRA$FNEXT &
    sleep $STR_DELAY
done

```

## 13.2 Příloha 2 – tabulky, zpracované výsledky

RTMP - datový tok 5 Mbit/s		
Počet připojených klientů (bez monitorovacího)	Průměrné využití CPU (load average)	využití RAM (RES) [MB]
40	0.41	1200
80	0.61	1300
120	1.69	1300
RTMP - datový tok 9 Mbit/s		
Počet připojených klientů (bez monitorovacího)	Průměrné využití CPU (load average)	využití RAM (RES) [MB]
80	2.28	1300
RTMP - datový tok 3 Mbit/s		
Počet připojených klientů (bez monitorovacího)	Průměrné využití CPU (load average)	využití RAM (RES) [MB]
40	0.21	351

Tab 12.1: RTMP – naměřené hodnoty pro různé datové toky

HDS - datový tok 1 Mbit/s		
Počet připojených klientů (bez monitorovacího)	Průměrné využití CPU (load average)	využití RAM (RES) [MB]
40	1.25	511
HDS - datový tok 3 Mbit/s		
Počet připojených klientů (bez monitorovacího)	Průměrné využití CPU (load average)	využití RAM (RES) [MB]
40	1.25	511
HDS - datový tok 5 Mbit/s		
Počet připojených klientů (bez monitorovacího)	Průměrné využití CPU (load average)	využití RAM (RES) [MB]
40	0.24	291

Tab 12.2: HDS - naměřené hodnoty pro různé datové toky

RTSP - datový tok 1 Mbit/s		
Počet připojených klientů (bez monitorovacího)	Průměrné využití CPU (load average)	využití RAM (RES) [MB]
40	1.25	511
80	1.84	469
120	1.96	336
RTSP - datový tok 3 Mbit/s		
Počet připojených klientů (bez monitorovacího)	Průměrné využití CPU (load average)	využití RAM (RES) [MB]
40	2.5	486
80	2.23	588
120	3.42	579
RTSP - datový tok 5 Mbit/s		
Počet připojených klientů (bez monitorovacího)	Průměrné využití CPU (load average)	využití RAM (RES) [MB]
40	2.13	277
80	2.87	404
120	3.42	681
RTSP - datový tok 9 Mbit/s		
Počet připojených klientů (bez monitorovacího)	Průměrné využití CPU (load average)	využití RAM (RES) [MB]
40	2.79	535
80	5.69	649

Tab. 12.3: RTSP – naměřené hodnoty pro různé datové toky

HDS - 40 klientů		
datový tok [Mbit/s]	Průměrné využití CPU (load average)	využití RAM (RES) [MB]
1	1.25	511
3	1.25	511
5	0.24	291

Tab. 12.4: HDS – naměřené hodnoty pro 40 klientů

RTSP - 40 klientů		
datový tok [Mbit/s]	průměrné využití CPU (load average)	využití RAM (RES) [MB]
1	1.25	511
3	2.5	486
5	2.13	277
9	2.79	535
RTSP - 80 klientů		
datový tok [Mbit/s]	Průměrné využití CPU (load average)	využití RAM (RES) [MB]
1	1.84	469
3	2.23	588
5	2.87	404
9	5.69	649
RTSP - 120 klientů		
datový tok [Mbit/s]	Průměrné využití CPU (load average)	využití RAM (RES) [MB]
1	1.96	336
3	3.42	579
5	3.42	681

Tab. 12.5: RTSP - naměřené hodnoty pro 40, 80 a 120 klientů

RTMP - 40 klientů		
datový tok [Mbit/s]	Průměrné využití CPU (load average)	využití paměti RAM (RES) [MB]
3	0.21	351
5	0.41	1200
RTMP - 80 klientů		
datový tok [Mbit/s]	Průměrné využití CPU (load average)	využití paměti RAM (RES) [MB]
5	0.61	1300
9	2.28	1300
RTMP - 120 klientů		
datový tok [Mbit/s]	Průměrné využití CPU (load average)	využití paměti RAM (RES) [MB]
5	1.69	1300

Tab. 12.6: RTMP – naměřené hodnoty pro 40, 80 a 120 klientů

## 13.3 Příloha 3 – DVD

Na přiloženém DVD se nachází:

- testovací script (příloha č.1)
- zdrojové video
- překomprimované videa do nižších datových toků
- výstupy z ostatních monitorovacích nástrojů
- upravené a popsané grafy pro lepší vizuální srovnání